Edited by
Andrii Shyshatskyi

INFORMATION AND CONTROL SYSTEMS: MODELLING AND OPTIMIZATIONS

Collective monograph

Technology Center

2024

UDC 004.8
I60

Published in 2024
by TECHNOLOGY CENTER PC®
Shatylova dacha str., 4, Kharkiv, Ukraine, 61165

I60 **Authors:**

Edited by **Andrii Shyshatskyi**
Olexander Litvinenko, Svitlana Kashkevich, Andrii Shyshatskyi, Oksana Dmytriieva, Serhii Neronov, Ganna Plekhova, Yevhen Zhyvylo, Nina Kuchuk, Oleksii Kuvshynov, Oleksandr Yefymenko, Andrii Veretnov, Heorhii Kuchuk, Viacheslav Davydov, Yurii Beketov, Illia Dmytriiev, Inna Shevchenko, Oleksandr Lytvynenko, Lyubov Shabanova-Kushnarenko, Nataliia Apenko

Information and control systems: modelling and optimizations: collective monograph. – Kharkiv: TECHNOLOGY CENTER PC, 2024. – 180 p.

The collective monograph "Information Management Systems: modeling and optimization" is the result of many years of painstaking work by researchers on solving optimization problems using the theory of artificial intelligence. Structurally and logically, the monograph is divided into sections structured in a certain thematic area of research using artificial intelligence methods. This work is not comprehensive and does not claim to be complete in this scientific and theoretical direction, which will be developed in the next part of the monograph.

The monograph will be useful for researchers involved in solving optimization problems, using the theory of artificial intelligence, and developing new (improving existing) approaches to solving complex technical problems in various fields of human activity.

The monograph is also useful for practitioners – designers, and developers implementing modern solutions in the field of information technology, engaged in the development of information, information-analytical, and automated systems.

Figures 14, Tables 39, References 169 items.

DOI: 10.15587/978-617-8360-04-7
ISBN 978-617-8360-04-7 (on-line)

Cite as: Shyshatskyi, A. (Ed.) (2024). Information and control systems: modelling and optimizations: collective monograph. Kharkiv: TECHNOLOGY CENTER PC, 180. doi: http://doi.org/10.15587/978-617-8360-04-7

9 786178 360047

AUTHORS

CHAPTER 1

OLEXANDER LITVINENKO
Doctor of Technical Sciences, Professor, Head of Department
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-8862-8032

SVITLANA KASHKEVICH
Senior Lecturer
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-4448-3839

ANDRII SHYSHATSKYI
Doctor of Technical Sciences, Senior Researcher,
Associate Professor
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0001-6731-6390

OKSANA DMYTRIIEVA
Doctor of Economic Sciences, Professor, Head of Department
Department of Economics and Entrepreneurship
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0001-9314-350X

SERHII NERONOV
Senior Lecturer
Computer Systems Department
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-2381-1271

GANNA PLEKHOVA
PhD, Associate Professor
Department of Informatics and Applied Mathematics
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-6912-6520

CHAPTER 2

SVITLANA KASHKEVICH
Senior Lecturer
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-4448-3839

ANDRII SHYSHATSKYI
Doctor of Technical Sciences, Senior Researcher,
Associate Professor
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0001-6731-6390

OKSANA DMYTRIIEVA
Doctor of Economic Sciences, Professor, Head of Department
Department of Economics and Entrepreneurship
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0001-9314-350X

YEVHEN ZHYVYLO
PhD, Associate Professor
Department of Computer and Information Technologies
and Systems
National University "Yuri Kondratyuk Poltava Polytechnic"
ORCID: https://orcid.org/0000-0003-4077-7853

GANNA PLEKHOVA
PhD, Associate Professor
Department of Computer Science and Applied Mathematics
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-6912-6520

SERHII NERONOV
Senior Lecturer
Computer Systems Department
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-2381-1271

CHAPTER 3

SVITLANA KASHKEVICH
Senior Lecturer
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-4448-3839

NINA KUCHUK
Doctor of Technical Sciences, Professor
Department of Computer Engineering and Programming
National Technical University "Kharkiv Polytechnic Institute"
ORCID: http://orcid.org/0000-0002-0784-1465

OLEKSII KUVSHYNOV
Doctor of Technical Sciences, Professor, Deputy Head of Center for Scientific Work
Military and Strategic Research Centre
The National Defense University of Ukraine
ORCID: http://orcid.org/0000-0003-2183-7224

GANNA PLEKHOVA
PhD, Associate Professor
Department of Informatics and Applied Mathematics
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-6912-6520

OLEKSANDR YEFYMENKO
PhD, Professor, Associate Professor
Department of Construction and Road-Building Machinery
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-0628-7893

ANDRII VERETNOV
PhD, Head of Department
Research Department
Central Scientific Research Institute of Armament and Military Equipment of Armed Forces of Ukraine
ORCID: https://orcid.org/0000-0003-0160-7325

CHAPTER 4

SVITLANA KASHKEVICH
Senior Lecturer
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-4448-3839

HEORHII KUCHUK
Doctor of Technical Sciences, Professor
Department of Computer Engineering and Programming
National Technical University "Kharkiv Polytechnic Institute"
ORCID: https://orcid.org/0000-0002-2862-438X

GANNA PLEKHOVA
PhD, Associate Professor
Department of Informatics and Applied Mathematics
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-6912-6520

VIACHESLAV DAVYDOV
Doctor of Technical Sciences, Associate Professor, Head of Department
Departments of Information Technologies
Science Entrepreneurship Technology University
ORCID: https://orcid.org/0000-0002-2976-8422

OLEKSANDR YEFYMENKO
PhD, Professor, Associate Professor
Department of Construction and Road-Building Machinery
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-0628-7893

YURII BEKETOV
PhD, Professor, Corresponding Member of Transport Academy of Ukraine
Department of Transport Technologies
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0002-0159-4950

CHAPTER 5

SVITLANA KASHKEVICH
Senior Lecturer
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0002-4448-3839

ILLIA DMYTRIIEV
Doctor of Economic Sciences, Professor
Department of Management
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0001-8693-3706

INNA SHEVCHENKO
Doctor of Economic Sciences, Professor
Department of Economics and Entrepreneurship
Kharkiv National Automobile and Highway University
ORCID: https://orcid.org/0000-0003-0758-9244

OLEKSANDR LYTVYNENKO
PhD, Senior Researcher
Research Department
Research Center
Military Institute of Taras Shevchenko National University of Kyiv
ORCID: http://orcid.org/0009-0000-6541-3621

LYUBOV SHABANOVA-KUSHNARENKO
PhD, Associate Professor
Department of Intelligent Computer Systems
National Technical University "Kharkiv Polytechnic Institute"
ORCID: https://orcid.org/0000-0002-2080-7173

NATALIIA APENKO
PhD, Associate Professor
Department of Computerized Control Systems
National Aviation University
ORCID: https://orcid.org/0000-0001-6891-0869

ABSTRACT

The high dynamism of the development of social processes and phenomena determines the formation of a new system of the worldview of mankind, the modification (change) of the hierarchy of needs and values, and challenges to the pace and quality of development.

Solving complex problems associated with meeting the requirements of our time requires the use of innovative scientific approaches. Today, the use of modern intellectual technologies, such as neural networks, deep learning, and artificial intelligence, is a prerequisite for the proactive development of all spheres of human activity: medicine, technology, business, environmental protection, education, transport and communication, etc. Thus, the intellectualization of technical and managerial systems can be considered one of the key foundations of the new paradigm of science and technology. The phrase "artificial intelligence systems" today is understandable to everyone. The context of this term is associated with such concepts as robotics, forecasting, processing of large information flows, expert systems, diagnostics, smart home or smart tools projects, cyberphysical space and cyberphysical systems, computer translation, etc.

There is a positive dynamics in the development and implementation of artificial intelligence elements in most types of software: mobile applications, information systems, electronic devices, etc.

This process of "intellectualization" allows us to talk about a gradual increase in the intelligence of modern computer systems capable of performing functions that are traditionally considered intellectual: understanding language, logical inference, using the accumulated knowledge, learning, pattern recognition, as well as learning and explaining their decisions.

The monograph provides methods for training artificial neural networks that have an adaptive structure and can evolve. They are set out in a separate section in the study. These methods are used by the authors in further studies to reduce errors that accumulate during the solution of optimization problems.

A separate section presents the issue of self-organization of information networks, using artificial intelligence methods. This study is aimed at solving the scientific and applied problem in terms of increasing the efficiency of self-organization of information networks at the first four levels of the model of interaction of open systems.

Separate sections include issues of evaluation and management of organizational and technical systems. These methods are based on metaheuristic algorithms. Assessment of the state of organizational and technical systems makes it possible to determine their state, taking into account the type of uncertainty about the available information, about their state, and in the future to develop adequate and reliable management decisions, taking into account the noise (distortion) of the data circulating in the organizational and technical system

The authors' research is supported by appropriate analytical expressions, graphic dependencies, and table values.

The monograph will be useful for researchers involved in solving optimization problems, using the theory of artificial intelligence, and developing new (improving existing) approaches to solving complex technical problems in various fields of human activity.

The monograph is also useful for practitioners – designers, developers implementing modern solutions in the field of information technology, engaged in the development of information, information and analytical, as well as automated systems to create new schemes and algorithms, their adaptation to non-stereotypical conditions of use, including for the implementation of artificial intelligence methods in the conditions of autonomous work, limitation of computing resources, remote control, etc.

KEYWORDS

Intelligent systems, decision support systems, artificial intelligence, artificial neural networks.

CIRCLE OF READERS AND SCOPE OF APPLICATION

The monograph will be useful for researchers involved in solving optimization problems, using the theory of artificial intelligence, and developing new (improving existing) approaches to solving complex technical problems in various fields of human activity.

The monograph is also useful for practitioners – designers, developers implementing modern solutions in the field of information technology, engaged in the development of information, information and analytical, as well as automated systems to create new schemes and algorithms, their adaptation to non-stereotypical conditions of use, including for the implementation of artificial intelligence methods in the conditions of autonomous work, limitation of computing resources, remote control, etc.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The high dynamism of the development of social processes and phenomena determines the formation of a new system of the worldview of mankind, the modification (change) of the hierarchy of needs and values, and challenges to the pace and quality of development.

Solving complex problems associated with meeting the requirements of our time requires the use of innovative scientific approaches. Today, the use of modern intellectual technologies, such as neural networks, deep learning, and artificial intelligence, is a prerequisite for the proactive development of all spheres of human activity: medicine, technology, business, environmental protection, education, transport and communication, etc. Thus, the intellectualization of technical and managerial systems can be considered one of the key foundations of the new paradigm of science and technology. The phrase "artificial intelligence systems" today is understandable to everyone. The context of this term is associated with such concepts as robotics, forecasting, processing of large information flows, expert systems, diagnostics, smart home or smart tools projects, cyber-physical space and cyberphysical systems, computer translation, etc.

There is a positive dynamics in the development and implementation of artificial intelligence elements in most types of software: mobile applications, information systems, electronic devices, etc.

This process of "intellectualization" allows us to talk about a gradual increase in the intelligence of modern computer systems capable of performing functions that are traditionally considered intellectual: understanding language, logical inference, using the accumulated knowledge, learning, pattern recognition, as well as learning and explaining their decisions.

In the first section of the monograph, a method of self-organization of information networks in conditions of destabilizing influences was proposed. The formalization of the task of managing the resources of the information network to the maximum bandwidth is carried out. In the context of this study, a comprehensive model of the functioning of information networks is proposed, as well as an algorithm for implementing the method of self-organization of information networks in conditions of destabilizing influences.

In the second section, the development of control methods based on bioinspired algorithms was carried out. The proposed methods are based on canonical bioinspired algorithms such as the flying squirrel algorithm, the goose flock algorithm, and the snake flock algorithm. The effectiveness of the proposed bioinspired algorithms was simulated, as well as their comparison with known ones.

The third section is the development of methods for assessing the state of complex technical systems using the theory of artificial intelligence. In this study, the following swarm algorithms were used as the basic ones: a flock of bats algorithm, an invasive weed algorithm, a flock of fish algorithm, as well as a flock of frogs algorithm. The advantages and disadvantages of each approach are determined, and the indicators for assessing the effectiveness of these algorithms with the appropriate justification are determined.

In the fourth section, the development of methods for training artificial neural networks of intelligent decision support systems was investigated. A method for training artificial neural networks, with evolving architecture, a method for training artificial neural networks Kohonen, with evolving architecture, as well as a method for training cascading artificial neural networks, with evolving architecture are proposed. The effectiveness of the proposed methods was evaluated, and their advantages and disadvantages were determined.

The fifth section of the monograph presents a scientific and methodological apparatus for increasing the efficiency of information processing using artificial intelligence. A mathematical formulation of the research problem was carried out with the help of a pack of wolves. A mathematical model of parametric optimization was developed based on an improved algorithm of fireflies in special-purpose information systems. The following scientific results were developed: a method of parametric optimization based on an improved algorithm of a pack of wolves; a method of parametric evaluation of the control object based on an improved firefly algorithm; methods of finding solutions using an improved locust flock algorithm; method of finding solutions using an improved algorithm for emperor penguins. The effectiveness of the proposed scientific results was evaluated, and the advantages and disadvantages of the proposed scientific results were determined.

DOI: 10.15587/978-617-8360-04-7.CH1

Olexander Litvinenko, Svitlana Kashkevich, Andrii Shyshatskyi,
Oksana Dmytriieva, Serhii Neronov, Ganna Plekhova

CHAPTER 1

THE METHOD OF SELF-ORGANIZATION OF INFORMATION NETWORKS IN THE CONDITIONS OF DESTABILIZING INFLUENCES

ABSTRACT

In this chapter of the research, a method of self-organization of information networks in conditions of destabilizing influences is proposed. The basis of this research is the theory of artificial intelligence, namely evolving artificial neural networks, basic genetic algorithm procedures and bio-inspired algorithms.

In the course of the research, the authors proposed:

– mathematical model of information conflict of information networks. The advantages of the specified model are due to take into account a greater number of destabilizing factors, compared to the known ones. The model takes into account in the complex deliberate interference of an additive and multiplicative nature, destabilizing factors due to the presence of cyber attacks;

– the method of self-organization of special purpose information networks. The novelty of the method consists in the use of additional and improved procedures that allow: to take into account the type of uncertainty and noisy data; to implement adaptive strategies for finding food sources; to combine individual swarm search strategies into a single strategy; to take into account the presence of a predator while choosing food sources; to take into account the available computing resources of the system while implementing self-organization of information networks; to change the search area by individual agents of the combined algorithm swarm; to change the speed of movement of agents of the combined algorithm swarm; to take into account the priority of searching for swarm agents of the combined algorithm; to carry out the initial display of individuals of the flock of the combined algorithm, to take into account the type of uncertainty; to carry out accurate training of individuals of the flock of the combined algorithm; to determine the best individuals of the flock of the combined algorithm using an advanced genetic algorithm;

– improved method of complex management of information network resources. The specified method allows: to determine the influence of destabilizing factors on the information network, to describe the information network of different architecture.

KEYWORDS

Information networks, self-organization, multi-agent systems, information conflict, reliability and adequacy.

1.1 COMPLEX MODEL OF FUNCTIONING OF INFORMATION NETWORKS

Currently, the method of the theory of information conflicts in the technical field is being formed as a natural development and synthesis of previously disconnected theories of electronic warfare (EW) and information security.

It should be noted that the concept of "information conflict" was introduced quite a long time ago to research the processes of antagonistic interaction of information networks associated with a violation of the availability, integrity and confidentiality of information [1].

In the majority of works on information conflict, it is considered in the context of the use of radio-electronic countermeasures in order to disrupt the functions of information support [1–4].

In general, an information conflict is characterized by its inherent hierarchical structure, corresponding to different levels of extracted (necessary) information and, accordingly, levels of extraction, collection and generalization of data about the conflicting parties. The network architecture of modern information networks is based on the seven-level model of the interaction of open systems OSI (The Open Systems Interconnection model), which provides for the independence of management functions by level [5–10].

A lower physical level of information conflict is the opposition of radio-electronic systems with their purposeful use of various types of electromagnetic radiation and effects on physical infrastructure in the interests of obtaining primary information about the characteristics and condition of the main objects of the opposite party or/and preventing the possibility of receiving such information by the other party [1].

At the same time, as shown in works [1–29], there is currently a retreat of the information conflict into the global telecommunications space. Available works [2–14] in the part of informational conflict within the framework of the duel "informational and management system – system of destabilizing influences", as a rule, consider the conflict at one level of functioning of these systems. At the same time, in the works [1, 3, 7, 13] it is noted that the information conflict has a complex hierarchical nature and can consist of many duel-game situations at different levels of the hierarchy.

The works [1, 2] present the development of the information conflict model in the direction of increasing the "multilevel" of the conflict and its alignment with the OSI model. These works suggest, together with the "classical" information conflict, to take into account new methods of influence due to the decomposition of the information conflict of information networks into separate conflict situations at each of the levels of the OSI model.

Thus, the new conceptual approach to information conflict modeling proposed in works [1, 2], on the one hand, organically develops existing works [3, 5, 11] in the part of multi-level information conflict of information networks and on the other hand, formalizes conflict interaction according to levels of the OSI reference model.

This conceptual model is a reference model of the interaction of conflicting systems CSI (Conflict System Interconnection Reference Model) formalizes objects and general approaches to the description of local information conflicts in information networks at each of the levels of the OSI model.

At the same time, taking into account the great contribution of these works to the development of the theory of the conflict of information networks and destabilizing factors, the following questions remain unresolved:

– the description of the conflict of information networks and destabilizing factors is limited to only one (several) levels without describing its specified impact on other levels of the OSI model;

– the time for collecting, processing and summarizing information for making management decisions by the information management system is not taken into account;

– a priori uncertainty about devices of destructive influence is not taken into account;

– the degree of data noise is not taken into account.

This fact determines the relevance of the conducted research. The generalized expression of the mathematical model of the information conflict of information networks under conditions of destabilizing influences is presented in expression (1.1):

$$\forall t \in \{1,\ldots,T,\ldots\} S_t = \left\{ s_i^{(t)} F_i \left(\left(\varphi_{1,i} \left(s_i^{(t-1)} \left(X_i, \Pi_i, U_i, A_i, \Omega_i, Y_i, Q_i, N_i, M_i \right) \right) \right) \cdot \iota_i \right) \cdot \chi_i, \; i = \overline{1,7}, \right. \quad (1.1)$$

where S is a multidimensional time series; $S_t = \left(s_1^{(t)}, s_2^{(t)}, \ldots, s_N^{(t)} \right)$ is a time slice of the state of the information network is presented in the form of a multidimensional time series at the t-th moment of time; $s_j^{(t)}$ is the value of the j-th component of the multidimensional time series at the t-th moment of time; L_j^i is the maximum value of the time delay of the i-th component relative to the j-th; φ_{ij} is the operator for taking into account the interaction between the i-th and j-th components of the multidimensional time series; F_i is the transformation for obtaining $s(t)$, $i=1,\ldots,N$; N is the number of components of a multidimensional time series; ι is the operator for taking into account the degree of awareness of the devices of information influence on the information network; χ is the operator for taking into account the degree of noise of data on the state of the information network.

From the expression (1.1), it is possible to conclude that the expression allows to describe the processes in the information network taking into account time delays. Delays are necessary for the collection, processing and generalization of information. An expression (1.1) also takes into account the degree of awareness of the devices of information influence and data noise, describes processes that have both quantitative and qualitative units of measurement.

Let's describe all the components of expression (1.1) in detail. Each l-th level of the information network solves its functional tasks. At the same time, the level itself is formed by a set of functionally interacting protocols $\Pi_l = \bigcup \pi_l$, which function in the parametric space of the environment $X_l = R_l \times \chi_l \times V_l$, where $\chi_l = \bigcup \chi_{l,\pi}$, $R_l = \bigcup R_{l,\pi}$, $V_l = \bigcup V_{l,\pi}$. Accordingly, each level has its own specific parameters of the natural environment χ_l, resources R_l, and influences V_l. At the same time, the effectiveness of the functioning of the lower levels directly determines the effectiveness of the functioning of the higher levels.

The information network consists of 7 levels that correspond to the OSI model, and each of the levels corresponds to its own set of protocols. Taking into account the formal description

of the protocol presented above, each level of the model in the hierarchical model of the information network will be described by the following parameters and mappings:

1. Input parameters of the l-th level: a set of generalized parameters of the environment $X_l = R_l \times \chi_l \times V_l$, where $\chi_l = \bigcup \chi_{l,\pi}$, $R_l = \bigcup R_{l,\pi}$, $V_l = \bigcup V_{l,\pi}$ in which a number of protocols $\Pi_l = \bigcup \pi_l$ function on the l-th level, which include: a set of parameters of the natural environment of the l-th level $\chi_l \left(\chi_l \subseteq \chi\right)$; a set of information network resources of the l-th level $R_l \left(R_l \subseteq R\right)$; a set of destructive influences V_l, which are implemented by the system of destructive influences on the l-th level and affecting the functioning of protocols $\Pi_l = \bigcup \pi_l$; a multitude of governing influences $U_l = <\{U_{l,\pi}\} \times T> \left(U_l \subseteq U\right)$ on protocols Π_l from the information network at the l-th level; a set of time points of the functioning of the information network T.

2. Transformations defining the general dynamic model of the l-th level of the information network:

– a set of algorithms A_l of the protocols Π_l at the l-th level of the information network:

$$A_l = \bigcup_{\Pi_l} A_{l,\pi}; \tag{1.2}$$

– a set of parameters Ω_l of the algorithms A_l of protocols at the l-th level of the information network:

$$\Omega_l = \bigcup_{\Pi_l} \Omega_{l,\pi}; \tag{1.3}$$

– an expression ψ_l, which describes the change of states s_l of l-th level of the information network:

$$s_l = \left\{s_\pi\right\} \bigcup \Theta_l = \psi_l \left(t_0, t, \left\{\pi\right\}, \left\{S_\pi\right\}, X_l, U_l, A_l, \Theta_l\right); \tag{1.4}$$

– at the same time, the state of the l-th level is defined as the union of the set of states of all protocols of this level $\{s_\pi\}$ and the state of functional-structural connections between protocols Θ_l;

– transformation f_l, which determines the initial indicators of service quality Q_l, which provide protocols Π_l at the l-th level of functioning:

$$Q_l = f_l \left(t, s_l, X_l, U_l, A_l\right); \tag{1.5}$$

– transformation γ_l, which defines the output parameters Y_l of the protocols Π_l at the l-th level:

$$Y_l = \gamma_l \left(t, s_l, X_l, U_l, A_l\right); \tag{1.6}$$

– transformation φ_l, which defines the parametric set of the environment X_{l+1} functioning of higher level protocols Π_l:

$$X_{l+1} = \varphi_l \left(s_\pi, t, X_l, U_l, A_l\right). \tag{1.7}$$

3. Output parameters: a set of output parameters of the l-th level Y_l: a set of service quality indicators Q_l, which are provided at the l-th level of functioning; a set of environment parameters for the functioning of higher-level protocols X_{l+1}; surveillance channel $N_l = \bigcup N_{l,\pi} = < Y_{l,N} \times Q_{l,N} >$ on the part of the information network in the interests of making decisions on the management of the information network; surveillance channel $M_l = \bigcup N_{l,\pi} = < Y_{l,M} \times Q_{l,M} >$ from the side of the system of destabilizing influences in the interests of making decisions about the expedient application of influences V_l.

Taking into account the above, the hierarchical model of the information network as a set of levels of the OSI model will be determined by the following parameters and transformations:

1. Input parameters of the information network: a set of generalized parameters of the environment $X = \bigcup X_l$, which include: a set of parameters of the natural environment set $\chi = \bigcup \chi_l$ of information network resources $R = \bigcup R_l$; multiple influences $V = \bigcup V_l$ implemented by the system of destabilizing influences at l-th levels of functioning; a set of controlling influences $U = \bigcup U_l$ from the side of the information network at the l-th levels of functioning; the set of time points of the functioning of T.

2. Transformations defining the dynamic model of the information network:

– a set of algorithms A of the information network:

$$A = \bigcup A_l; \tag{1.8}$$

– the set of parameters Ω of algorithms A of the information network:

$$\Omega = \bigcup \Omega_l; \tag{1.9}$$

– the set of Π protocols of the information network:

$$\Pi = \bigcup \Pi_l; \tag{1.10}$$

– the set ψ specifying the change of states S of the information network:

$$S = \{S_l\} \bigcup \Theta = \psi\left(t_0, t, \{S_l\}, X, U, A, Z, E, K, \Lambda, \Theta\right), \tag{1.11}$$

where $Z = \{z_i\}$ is a set of information network elements; $E = \{e_i\}$ is a set of channel-forming devices in the information network; $K = \{k_{ij}\}$ is a set of channels connecting nodal elements of the information network;

– a set of transformations f that determine the initial indicators of the quality of service Q of the information network:

$$Q = f\left(t, S, X, U, A\right),\ f = \bigcup f_l,\ Q = \bigcup Q_l; \tag{1.12}$$

– a set of transformations γ that determine the initial parameters Y of the information network:

$$Y = \gamma(t, S, X, U, A),\ \gamma = \bigcup \gamma_l,\ Y = \bigcup Y_l; \tag{1.13}$$

– a set of transformations φ that determine the hierarchical relationships between the levels of the information network:

$$\varphi = \bigcup \varphi_l. \tag{1.14}$$

In fact, the set of transformations φ determines the difference between this model and the classic OSI model, which does not explicitly specify inter-level relationships.

3. Output parameters: a set of output parameters of the information network Y; final output parameters of the information network Y_7; a set of service quality indicators Q of the information network; surveillance channel $N = \bigcup N_l = <Y_N \times Q_N>$ on the part of the information network in the interests of making decisions on the management of the information network; surveillance channel $M = \bigcup M_l = <Y_M \times Q_M>$ from the side of the system of destabilizing influences in the interests of decision making regarding the application of multi-level influences V.

In the general case, the processes of functioning of the information network are given by transformations (1.10)–(1.14). The basis of the information network is a spatially distributed transport network, the state of its elements determines the state S of the information network as a whole:

$$S = \psi\left(t_0, t, \{S_l\}, X, U, A, Z, E, K, \Lambda, \Theta\right), \tag{1.15}$$

where $E = \{e_i\}$ is a set of channel-forming devices installed on the nodal elements of the information network, while each devices supports a set of protocols π_i, so $e_i = <\{\pi_i\}>$; $Z = \{z_i\}$ is a set of nodal elements of the network, each nodal element contains a set of channel-forming devices e_i, so $z_i = <\{e_i\}, \{\pi_i\}, \{k_{ij}\}>$; $K = \{k_{ij}\}$ is a set of communication channels connecting nodal elements of the information network; Λ is the distribution matrix of information flows between subscribers and nodal elements of the information network (attraction matrix).

The $Z \times K$ parameters actually set the graph of the information network and individual actions $V_{l,\pi}$ due to malfunctioning or blocking of the protocols used in the communication channel $k_{ij} = <z_i, z_j, e_i, e_j, \pi_{l,k}>$, as well as in the node $z_i = <\{e_i\}, \{\pi_i\}, \{k_{ij}\}>$, lead to a change in topology.

The process of functioning of the information network is determined by the transformation $\gamma : <t, S, X, U, A> \rightarrow Y$, given by expression (1.13).

In the process of functioning of the information network, messages Y_7 are delivered with quality indicators Q corresponding to the intended purpose of the information network depending on its current state S and the set of input influences X and management U.

The quality of performance of the functions according to the target purpose of the information network is determined by Q indicators, which depend on the set of used protocols Π, which function

according to algorithms A and their parameters Ω, and also depend on the parameters of the environment $X = R \times \chi \times V$.

The information network functions effectively if its quality indicators are of the upper (applied) level Q_7, obtained as a result of convolution $Q_1 \to Q_2 \to \ldots \to Q_i \to \ldots \to Q_7$, have a value not lower than the required value, so the criterion is fulfilled:

$$\forall \left\{ q_7 \geq q_7^{neces} \right\} \in Q_7. \tag{1.16}$$

At the same time, the four lower levels are of particular importance: physical, channel, network and transport, which correspond to the telecommunications equipment of the information network and are more vulnerable to intentional influences. Therefore, as a special case, it is possible to consider the information network as a spatially distributed transport network limited by the four lower levels of OSI. In this case, criterion (1.15) can be represented as:

$$\forall \left\{ q_4 \geq q_4^{neces} \right\} \in Q_4. \tag{1.17}$$

Let's consider information conflict in more detail by levels of influence:

a) impact only on the physical level (for example, RES);

b) influence only at the network level (for example, DDOS attacks);

c) assessment of the effects of single-level influence on the physical level (for example, the effects of RES lead to a loss of reliability of reception, which cannot be corrected by protocols of the physical and channel level, but due to rerouting at the network level, the effects of the influence are corrected);

d) multi-level complex influence (by analogy with the previous case, only the additional use of information influence at the network level leads to unstable functioning of routing protocols and the delivery of packets is impossible).

At the same time, the most likely scenario is the use of non-critical dynamic actions at the lower levels $\left(Q_1\left(V_1\right) > Q_1^{neces}\right)$, the effect of which is reflected on higher levels $\left(V_1 \to X_1 \to \Pi_1 \to Y_1 \to \to X_2 \to \Pi_2 \to \ldots\right)$ and brings the information network into an extremely stable state.

As it is shown in work [18], the effectiveness of managing a multi-level system, which is an information network, depends on the relationship between the global and local aims of functioning at each of the levels. In this regard, inter-level inconsistency (conflict) between locally adopted decisions may arise. In general, two types of conflicts arise in a multi-level system [18]: inter-level and intra-level.

As shown above, an information network is effective if its quality indicators are of the upper (applied) level Q_7, obtained as a result of convolution $Q_1 \to Q_2 \to \ldots \to Q_i \to \ldots \to Q_7$, have a value not lower than the required value, that is, the criterion $\forall\{q_7 \geq q_7^{neces}\} \in Q_7$ is fulfilled.

By analogy, it is possible to introduce the concept of effective functioning conditions – conditions $X_{ef} = \bigcup X_l \left(X_l = < R_l \times \chi_l \times V_l >\right)$, which in the management of the information network

$U = \bigcup U_l$ ensure the fulfillment of the functioning efficiency criterion $\forall\{q_7 \geq q_7^{neces}\}$. The existence of such effective conditions determines areas of ineffective actions – that is, impacts $V_{inef} = \bigcup V_l$, which under management $U = \bigcup U_l$ are not able to bring the information network beyond the limits of effective functioning, so $Q\left(U, X_{ef}\left(V_{inef}\right)\right) \geq Q^{nonord}$. In other words, the effective conditions of the functioning of the information network correspond to the condition of ineffective actions.

Currently, the classic principle of independence of the information network at different levels of its functioning is accepted in the OSI model. The classical principle of multi-level functioning according to the OSI model is based on two provisions:

1. Protocols of higher levels and their management system should ensure correction of ineffective initial parameters (initial conditions) of lower levels $Y_{l-1\ inef} \rightarrow Y_{l\ ef}$:

$$\varphi_l : Y_{l-1\ inef} \xrightarrow{\Pi_l} \left[\begin{array}{l} X_{l\ ef} \\ X_{l\ inef} \end{array} \right., \tag{1.18}$$

where $X_{l\ ef}$ corresponds to the case when protocols Π_l correct inefficient conditions of functioning of the previous level; $X_{l\ inef}$ – when they don't fix it.

Within $X_{l\ inef} = \bigcup\limits_{\Pi_l} X_{l,\pi\ inef}$, $X_{l\ ef} = \bigcup\limits_{\Pi_l} X_{l,\pi\ ef}$ the end $X_l = X_{l\ ef} \bigcup X_{l\ inef}$.

2. Protocols Π_l at their level of functioning functionally independent of lower levels Π_{l-1} and must ensure the necessary efficiency of functioning:

$$f_l : < t, s_l, X_{l\ ef}, U_l, A_l > \xrightarrow{\Pi_l} Q_l \middle| Q_l \geq Q_l^{nonord}. \tag{1.19}$$

An expression (1.18) allows to draw a conclusion that at first glance seems logical, if the initial parameters of the lower-level protocols are effective, then the operating conditions of the higher-level protocol are also effective:

$$Y_{l-1\ ef} \Rightarrow Y_{l\ ef}, \tag{1.20}$$

from where, extending this conclusion to all levels of the information network, it is possible to obtain that the effective conditions of the *l*-th level are the cause of the effective conditions of all subsequent levels:

$$X_{1\ ef} \Rightarrow Y_{1\ ef} \Rightarrow \ldots \Rightarrow Y_{l-1\ ef} \Rightarrow X_{l\ ef} \Rightarrow \ldots \Rightarrow Y_{7\ ef}. \tag{1.21}$$

At the same time, it is known from practice that there are modes of operation of protocols that are ineffective, but at the same time the conditions of effective operation of the 1st level are observed. Thus, Corollary (1.21) does not hold. As a rule, the explanation of the existence of such conditions lies in the field of independence of the functional levels of the information network and, allegedly, the impossibility of forming separate conditions of the current level from the conditions of functioning of the previous level.

For the deliberate implementation of the above conditions of inefficient functioning of the information network, three basic options for influence can be proposed:

1) influence at a lower level V_i, which is inefficient at the i-th level (that is, does not lead to a decrease $Q_i < Q_i^{nonord}$), but due to influence:

$$V_{i\ inef} \to X_{i\ ef} \to Y_{i\ ef} \to \ldots \to X_{j\ inef} \to Y_{j\ inef}$$

to the higher j-th level of the information network forms its inefficient environment $X_{j\ inef}$, which leads to a critical drop in the quality of functioning of this level;

2) a set of influences $V = \bigcup V_i$ at different levels which are individually ineffective, but due to their influence on higher levels creates ineffective environmental conditions $X_{j\ inef}$ at the highest j-th level:

$$X_{1\ ef} \to Y_{1\ ef} \to \ldots \to \underset{\uparrow}{\overset{V_{i-1\ inef}}{\downarrow}}\!\!\!\!X_{i-1\ ef} \to Y_{i-1\ ef} \to \overset{V_{i\ inef}}{\underset{}{\downarrow}}\!\!\!X_{i\ ef} \to Y_{i\ ef} \ldots \to X_{j\ inef} \to Y_{j\ inef};$$

3) single influence V_i or a combination $V = \bigcup V_i$ of them at different levels, which lead to the emergence and development of intrasystem conflicts in the information network both within a separate level and between levels of functioning.

1.2 THE METHOD OF SELF-ORGANIZATION OF INFORMATION NETWORKS IN CONDITIONS OF DESTABILIZING INFLUENCES

Optimization is a complex process of identifying multiple solutions for a variety of functions. Many calculation tasks today belong precisely to optimization tasks [1–3]. While solving optimization tasks, decision variables are determined in such a way that information networks work at their best point (mode) based on the optimization criteria determined.

The problems of optimization of information networks are discontinuous, undifferentiated and also multimodal. Thus, it is impractical to use classical gradient deterministic algorithms [4–6] to solve the tasks of self-organization of information networks.

To overcome the shortcomings of classical optimization algorithms for solving problems of self-organization of information networks, a significant number of stochastic optimization algorithms, known as metaheuristic algorithms, were created [7–9].

Swarm intelligence algorithms (swarm algorithms) are one of the types of algorithms for stochastic optimization of information networks. Swarm intelligence algorithms are based on swarm movement and simulate interactions between the swarm and its environment to improve knowledge of the environment, such as new food sources. The most famous swarm algorithms are the particle swarm optimization algorithm, the artificial bee colony algorithm, the ant colony optimization algorithm, the wolf pack optimization algorithm and the sparrow flock algorithm [11–17].

Unfortunately, most of the basic metaheuristic algorithms mentioned above are not able to balance exploration and use, which leads to unsatisfactory performance for real tasks of self-organization of information networks.

This motivates the implementation of various strategies to improve the speed, convergence and accuracy of metaheuristic algorithms. One of the options for increasing the efficiency of decision making using metaheuristic algorithms is their combination, thus, adding the basic procedures of one algorithm to another.

Taking into account the above, an urgent scientific task is to develop a method of self-organization of information networks under the conditions of complex influence of destabilizing factors using artificial intelligence, which would allow to increase the efficiency of decisions made regarding the management of parameters of self-organization of information networks with a given reliability.

The analysis of works [9–21] showed that the common shortcomings of the above-mentioned researches are: the lack of the possibility of forming a hierarchical system of indicators regarding the process of self-organization of information networks; the lack of consideration of the computing resources of the system, which carry out the analysis of the process of self-organization of information networks; lack of mechanisms for adjusting the system of indicators during the evaluation of the process of self-organization of information networks; the lack of consideration of the type of uncertainty and noise of data on the process of self-organization of information networks, which creates corresponding errors in assessing their real state; lack of mechanisms for deep learning of knowledge bases that characterize the process of self-organization of information networks; high computational complexity while calculating the process of self-organization of information networks; the lack of consideration of computing (hardware) resources available in the system that evaluates the process of self-organization of information networks; the absence of the possibility of determining the priority of finding a solution, in a certain direction, regarding the state of the process of self-organization of information networks.

The purpose of the research is to develop a method of self-organization of information networks under the conditions of complex influence of destabilizing factors. This will allow to increase the speed of assessment of the process of self-organization of information networks with a given reliability and the development of subsequent management decisions. This will make it possible to develop software for intelligent decision making support systems that analyze the state of complex dynamic objects.

In order to detail the process of self-organization of information networks, it is possible to determine the parameters by which self-organization is carried out according to the levels of interaction of open systems (**Table 1.1**).

Table 1.1 Approximate relationship between parameters and control variables according to the levels of the OSI model

OSI layer	Management objects	Main optimization parameters	Controlling influence of the node
Physical	A channel within the limits of connectivity with neighboring nodes	Bandwidth, channel transmission time, battery power consumption, transmission power, etc.	Power (direction) of transmission, type of modulation, type of correction code, parameters, etc.
Channel	Channels within connectivity with neighboring nodes	Bandwidth and transmission time in the channel, battery energy consumption, amount of service information, etc.	Channel level exchange algorithms: deterministic, random, hybrid; sizes of packages and receipts
Network	One or more transmission routes	The amount of service information, route parameters (time of construction and existence, quantity, throughput, delivery time, battery energy consumption, etc.)	Network level exchange algorithms: table, probe, hybrid, wave asymmetric, hierarchical, etc. Topology control algorithms
Transport	Information direction of communication	Bandwidth, time and variation of its transmission in direction	Queue management algorithms. Overload window size, timeout time, etc.
Applied	Node, neighbor nodes, network zone, entire network	Bandwidth, transmission time and time variation, battery energy consumption, transmission security	Algorithms (protocols) of application-level information exchange, coordination and intellectualization by OSI levels

As can be seen from **Table 1.1**, the term self-organization of information networks mainly includes parameters 1–4 levels of interaction of open systems. Thus, in this study, let's consider the specified levels of interaction of open systems and the main parameters used in them.

The combined algorithm consists of three main steps (procedures): research, exploitation and transitional processes from research to exploitation. The method of self-organization of information networks in conditions of destabilizing factors consists of the following sequence of steps:

Step 1. *Input of initial data*. At this stage, the initial data available on the self-organizing information network are entered.

Step 2. *Exposure of individuals of the combined flock on the search plane*.

All the listed individuals, namely hawks and coots, form a population of the combined algorithm, which can be modeled from a mathematical point of view using a matrix according to equation. The presentation of individuals of the combined algorithm is carried out taking into account the uncertainty regarding the self-organizing information network and the initialization of the basic model of its state [2, 18, 20] (1.22):

$$X=\begin{bmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_i \\ \cdot \\ \cdot \\ \cdot \\ X_N \end{bmatrix}_{N\times m}=\begin{bmatrix} x_{1,1}\times\iota_{1,1} & \cdots & x_{1,d}\times\iota_{1,d} & \cdots & x_{1,m}\times\iota_{1,m} \\ \vdots & & \vdots & & \vdots \\ x_{i,1}\times\iota_{i,1} & \cdots & x_{i,d}\times\iota_{i,d} & \cdots & x_{i,m}\times\iota_{i,m} \\ \vdots & & \vdots & & \vdots \\ x_{N,1}\times\iota_{N,1} & \cdots & x_{N,d}\times\iota_{N,d} & \cdots & x_{N,m}\times\iota_{N,m} \end{bmatrix}_{N\times m}. \tag{1.22}$$

The main position of the flock of the combined algorithm is initialized at the beginning of the algorithm execution using equation (1.23):

$$x_{i,d}=lb_d+r\cdot\left(ub_d-lb_d\right), \tag{1.23}$$

where X is the population matrix of individuals of the combined algorithm; X_i is the i-th member of individuals of the combined algorithm (solution candidate); $x_{i,d}$ is the d-th dimension in the search space (decision variable); N is the number of individuals of the combined algorithm; m is the number of decision variables; r is a random number in the interval [0,1]; lb_d and ub_d are the lower and upper bounds of the decision variables, respectively.

Since the position of each individual of the combined algorithm in the solution space of the task represents a solution to the problem, the value of the objective function can be estimated accordingly for each individual of the combined algorithm. Accordingly, the set of estimated values for the objective function can be written according to equation (1.24):

$$F=\begin{bmatrix} F_1 \\ \cdot \\ \cdot \\ \cdot \\ F_i \\ \cdot \\ \cdot \\ \cdot \\ F_N \end{bmatrix}_{N\times 1}=\begin{bmatrix} F\left(X_1\right) \\ \cdot \\ \cdot \\ \cdot \\ F\left(X_i\right) \\ \cdot \\ \cdot \\ \cdot \\ F\left(X_N\right) \end{bmatrix}_{N\times 1}, \tag{1.24}$$

where F is the vector of the estimated objective function; F_i is the estimated objective function based on the i-th member of the flock of the combined algorithm.

The estimated values of the objective function provide valuable information about the quality of the solution options proposed by the swarm members of the combined population. The best value

obtained for the objective function corresponds to the best member of the combined algorithm swarm (the best possible solution) and the worst value obtained for the objective function corresponds to the worst member of the combined algorithm swarm (the worst possible solution).

Since at each iteration the position of the swarm of individuals of the combined algorithm in the problem solution space is updated, the best member must also be updated based on the comparison of the updated values for the objective function. At the end of the algorithm implementation, the position of the best member of the flock of the combined algorithm, obtained during the iterations of the algorithm, is presented as a solution to the task.

Step 3. Numbering of individuals in the flock of the combined algorithm, i, $i \in [0, S]$. At this stage, each individual of the flock of the combined algorithm is assigned a serial number.

Step 4. Determination of the initial speed of individuals of the flock of the combined algorithm. The initial speed v_0 of each individual of the combined algorithm is determined by the following expression:

$$v_i = \left(v_1, v_2 \ldots v_S\right),\ v_i = v_0. \quad (1.25)$$

In the planning of the proposed approach of the combined algorithm, the position of the population members in the problem-solving space is updated based on the simulation of the hunting strategy of the individuals of the combined flock in the wild.

Step 5. Preliminary assessment of the search (feeding) area by individuals of the combined flock. The diet of individuals of a combined flock is diverse, for a flock of hawks it is food of animal origin and for a flock of coots it is precisely food of plant origin. they eat hares, birds. Therefore, it is advisable to sort the quality of food. The choice of the place of feeding is carried out taking into account the degree of noise of the initial data, which is proposed in the work [46].

Step 6. Classification of food sources for combined swarm agents.

The location of the best food source (minimum suitability) is considered to be hares for hawk agents, small fish for coot agents (*FSht*), locations from the next three food sources are small birds (hawk agents), duckweed (coot agents) (*FSat*) and the rest are considered regular food (*FSnt*):

$$FSht = FS(sorte\_index(1)), \quad (1.26)$$

$$FSat(1:3) = \mathrm{FS}(sorte\_index(2:4)), \quad (1.27)$$

$$FSnt(1:NP\text{-}4) = FS(sorte\_index(5:\ \mathrm{NP})). \quad (1.28)$$

Step 7. Sorting the best individuals of the flock of the combined algorithm. The selection of the best individuals of the flock of the combined algorithm is carried out using the improved genetic algorithm proposed in work [23]. While searching for food, the strongest individuals of the combined algorithm flock with the largest sizes direct another individual from the combined flock in the group to search for food. This search behavior of combined swarm agents leads to different scanning areas of the search space, which improves the research ability of agents in global search.

Steps 1–7, 10–15 are common to all individuals of the combined algorithm. The remaining procedures are unique for each of the swarm optimization algorithms.

Step 8. Procedure for optimizing a flock of hawk agents.

Step 8.1. Execution of the intelligence procedure of the algorithm of a flock of hawk agents.

The formula for updating the hawks' position at this stage is as follows:

$$X(t+1)=\begin{cases} X_{vip}(t)-r_1\cdot|X_{vip}(t)-2r_2X(t)|,\ q\geq 0.5; \\ \left(X_{prey}(t)-X_m(t)\right)-r_3\left(LB+r_4(UB-LB)\right),\ q<0.5, \end{cases} \tag{1.29}$$

where $X(t)$ and $X(t+1)$ represent the position vector of hawk agents in the current and next iterations; t represents the current number of iterations; X_{prey} is the position of the sacrifice, which is also considered the optimal solution; $X_{vip}(t)$ is the position vector of a random individual from a flock of hawk agents in the current population; r_1, r_2, r_3, r_4 and q are random numbers between [0, 1]; LB and UB are the lower and upper limits of variables; $Xm(t)$ is the average position of all hawk agents in the population, which is calculated as follows:

$$X_m(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t), \tag{1.30}$$

where N is the size of the population; $X_i(t)$ is the vector of the current position of the i-th hawk agent from the flock.

Step 8.2. Research phase for the hawk-agent swarm algorithm.

In the algorithm, the escape energy of the victim E causes the algorithm to switch between global exploration and local exploitation phases. The energy of the prey gradually decreases during the escape process, which can be simulated as in equation (1.31):

$$E=2E_0\left(\frac{T-t}{T}\right), \tag{1.31}$$

where E_0 is a random number between [1, 1] representing the initial energy state of the prey, T is the maximum number of iterations. When $|E|\geq 1$, the hawk agent will continue to locate prey in the target area defined as the exploration phase. In the case of $|E|<1$, the hawk agent will start hunting the prey found in the previous stage and enter the exploitation stage.

In the operation phase, there are four possible strategies, including soft siege, hard siege, soft siege with gradual rapid dives and hard siege with gradual rapid dives to simulate the process of a hawk attacking its prey. r represents the probability of whether the victim will be able to escape the danger before a hawk attack, which is a random number between [0, 1]. If $r<0.5$ means that the prey successfully passed through the dangerous situation, $r\geq 0.5$ means a case of unsuccessful escape. Different combinations of r-value and escape energy E correspond to different hunting strategies. When $|E|<0.5$, a tough siege is being carried out. Otherwise, a soft siege is conducted.

Step 8.3. Execution of a soft siege strategy by individuals from a flock of hawk agents.

A soft siege is performed when $r \geq 0.5$ and $|E| \geq 0.5$. At this stage, the position of the hawk agent is updated as follows:

$$X(t+1) = \Delta X(t) - E\left|JX_{prey}(t) - X(t)\right|, \quad (1.32)$$

$$\Delta X(t) = X_{prey}(t) - X(t), \quad (1.33)$$

$$J = 2(1 - r_5), \quad (1.34)$$

where $\Delta X(t)$ is the distance between the position of the hawk agent and the victim; r_5 is a random number between [0, 1]; J is a random prey jump intensity.

Step 8.4. Execution of a strategy of hard siege by hawk agents.

A hawk agent will take a tough siege when $r \geq 0.5$ and $|E| < 0.5$. The mathematical description of such behavior can be presented as follows:

$$X(t+1) = X_{prey}(t) - E\,|\,\Delta X(t)\,|. \quad (1.35)$$

Step 8.5. Executing a soft siege strategy with gradual rapid dives for a flock of hawk agents.

When $r < 0{,}5$ and $|E| \geq 0.5$, the hawk agent will take a soft siege with gradual rapid dives. Levi's flight is integrated into the search procedure of the hawk flock algorithm and the mathematical model of the behavior described above is as follows:

$$Y = X_{prey}(t) - E\left|JX_{prey}(t) - X(t)\right|, \quad (1.36)$$

$$Z = Y + S \times LF(D), \quad (1.37)$$

$$X(t+1) = \begin{cases} Y, \text{ if } F(Y) < F(X(t)); \\ Z, \text{ if } F(Z) < F(X(t)), \end{cases} \quad (1.38)$$

where D is the dimension of the problem; S is a random vector whose size is $1 \times D$; $F(\cdot)$ is the objective function.

Only the best position between Y and Z is selected as the next position. $LF(\cdot)$ is the Levy flight function, which is calculated as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \quad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma(1+\beta) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta}, \quad (1.39)$$

where u and v are two random numbers between [0, 1]; β is a constant with a fixed value of 1.5; $\Gamma(\cdot)$ is the gamma function.

Step 8.6. Execution of a hard siege strategy of hawk agents with gradual rapid dives.

When $r < 0.5$ and $|E| < 0.5$, the hawk agent will perform a tight siege to get close to the prey and then launch a surprise attack. The mathematical model of this behavior is written as follows:

$$Y = X_{prey}(t) - E\left|JX_{prey}(t) - X_m(t)\right|, \tag{1.40}$$

$$Z = Y + S \times LF(D), \tag{1.41}$$

$$X(t+1) = \begin{cases} Y, \text{ if } F(Y) < F(X(t)); \\ Z, \text{ if } F(Z) < F(X(t)), \end{cases} \tag{1.42}$$

where $X_m(t)$ is calculated using equation (1.30). Only the best position between Y and Z is selected as the next position.

Step 9. Execution of the coot herd optimization algorithm.

Coot flock optimization is a population-based, gradient-free optimization method that simulates the collective behavior of American coots (a small waterfowl) on the water surface. This algorithm implements four different irregular and regular motions: random motion, chain motion, position adjustment based on group leaders and leader motion.

Coots usually live in a group and create a chain structure to move towards a target area (food). At the front of the flock are several coots, also known as flock leaders, who direct the direction and take responsibility for the entire flock. Therefore, according to the coot's habits, the original population is divided into two parts: coot-leader and coot-follower. If N is the population size, then the number of coot leaders is calculated as a percentage of the total population equal to L and the other members $(N - L)$ are considered coot followers. It is noted that all leaders are chosen randomly from the population. Then the mentioned four movements are performed.

Step 9.1. Random movement of coot agents.

At this stage, the random position Q is determined using equation (1.43). Coot followers move to this random position to explore different parts of the search domain:

$$Q = \text{rand}(1, D) \cdot (UB - LB) + LB, \tag{1.43}$$

where D is the dimension of the problem; LB and UB are the lower and upper limits of the variables. The random motion gives the algorithm better global search efficiency and enhances the algorithm's ability to deviate from the local optimum. The new position of the coot is updated as follows:

$$X_i(t+1) = X_i(t) + A \times r_6 \times (Q - X_i(t)), \tag{1.44}$$

where $X_i(t+1)$ is the position of the *i*-th follower in the next iteration t, r_6 is a random number in the range [0, 1] and the parameter A is calculated according to equation (1.45):

$$A = 1 - \frac{t}{T}, \tag{1.45}$$

where t is the number of current iterations and T is the maximum number of iterations.

Step 9.2. Chain movement of coot agents.

In the algorithm of a flock of coot agents, the average position of two people is used to perform chain movements. The new position of the follower coot is calculated as follows:

$$X_i(t+1) = \frac{1}{2} \times \left(X_{i-1}(t) + X_i(t)\right), \tag{1.46}$$

where $X_{i-1}(t)$ is the position of the (i–1)-th follower of the coot agent in the current iteration t.

Step 9.3. Position adjustment based on group leaders.

As a general rule, the whole group is led by one of the leaders of the group in front and all coots that remain must change their position based on the leaders and move towards them. However, a serious problem to be solved is that each coot must update its position according to the leader, using equation (1.47) designed to select the leader as follows:

$$k = 1 + (i \bmod L), \tag{1.47}$$

where i is the index of the current follower; L is the number of leaders; and k is the index number of the leader.

The next position of the coot follower based on the selected leader k is calculated using equation (1.48):

$$X_i(t+1) = Leader\,X_k(t) + 2 \times r_7 \times \cos(2R\pi) \times \left(Leader\,X_k(t) - X_i(t)\right), \tag{1.48}$$

where $Leader\ X_k(t)$ is the position of the selected leader; r_7 is a random number in the interval [0, 1] and R is a random number in the interval [–1, 1].

Step 9.4. Leadership movement of coot agents.

The group must be oriented to the optimal territory, so in some cases leaders have to leave the current optimal position in search of a better one. The formula for updating the leader's position is written as follows:

$$Leader\,X_i(t+1) = \\ = \begin{cases} B \times r_8 \times \cos(2R\pi) \times \left(gBest(t) - Leader\,X_i(t)\right) + gBest(t),\ r_9 < 0.5; \\ B \times r_8 \times \cos(2R\pi) \times \left(gBest(t) - Leader\,X_i(t)\right) - gBest(t),\ r_9 \geq 0.5, \end{cases} \tag{1.49}$$

In equation (1.49), *gBest* is the current optimal position; r_8 and r_9 are random numbers in the interval [0, 1] and R is a random number in the interval [1, 1]. In r_8 generates more significant stochastic movement to help the algorithm eliminate local optimal solutions. Cos($2R\pi$) is for finding the best person with different radius to get the top position. The value of B is calculated using equation (50):

$$B = 2 - t \times \left(\frac{1}{T}\right), \tag{1.50}$$

where t is the number of current iterations and T is the maximum number of iterations.

Step 10. *Combining individual optimization algorithms into a mixed one*.

To combine different types of natural optimization algorithms, an ensemble mutation strategy is used, which can generate various individuals to improve the global search capabilities of the hybrid algorithm, which is written as follows:

$$V_{i1} = \begin{cases} X_{R1} + F_1 \times \left(X_{R2} - X_{R3}\right), \; r_{10} < C_1; \\ X_i, \; r_{10} \geq C_1, \end{cases} \tag{1.51}$$

$$V_{i2} = \begin{cases} X_{R4} + F_2 \times \left(X_{R5} - X_{R6}\right) + F_2 \times \left(X_{R7} - X_{R8}\right), \; r_{11} < C_2; \\ X_i, \; r_{11} \geq C_2, \end{cases} \tag{1.52}$$

$$V_{i3} = \begin{cases} X_i + F_3 \times \left(X_{R9} - X_i\right) + F_3 \times \left(X_{R10} - X_{R11}\right), \; r_{12} < C_3; \\ X_i, \; r_{12} \geq C_3, \end{cases} \tag{1.53}$$

where V_{i1}, V_{i2} and V_{i3} are newly generated mutant positions of the i-th search agent; $R_1 \sim R_{11}$ are different integer indicators in the range [1, N]; F_1, F_2 and F_3 are scale factors with values of 1.0, 0.8, and 1.0, respectively; $r_{10} \sim r_{12}$ are random numbers in the range [0,1]. In addition, the parameters C_1, C_2 and C_3 are equal to 0.1, 0.2 and 0.9, which indicates the speed of the crossover.

After generating candidate mutant positions V_{i1}, V_{i2} and V_{i3}, the best position V_i with the lowest fitness value will be selected to compare with the fitness of the original position X_i and then the best position will be saved as a new X_i to participate in the next iteration calculation. These processes can be described using equation (1.54):

$$X_i = \begin{cases} V_i, \text{ if } F\left(V_i\right) < F\left(X_i\right); \\ X_i, \text{ otherwise}, \end{cases} \tag{1.54}$$

where $F(\cdot)$ is the cost function.

Step 11. *Checking the presence of predator agents of the combined flock*. At this stage, agents check for predators. If there are predators, go to Step 12. If there are no predators, go to Step 11.

Step 12. *Escape and struggle with predators of combined flock agents*. The strategy of escaping and fighting these predators causes the combined algorithm agents to change their position near the position they are at. Simulating this natural behavior of the combined algorithm agents

improves the power of using the combined algorithm in local search in the problem-solving space around potential solutions. Since this process occurs near the position of each combined swarm agent, it is assumed that this range of agent position change occurs in a corresponding zone centered on each combined swarm agent with a certain radius. In the initial iterations of the algorithm, priority is given to a global search to identify the optimal region in the search space, the radius of this environment is considered variable. First, the highest value is set, and then it becomes smaller during the iterations of the algorithm. For this reason, local lower/upper bounds were used to create a variable radius with iteration of the algorithm. To model this phenomenon, a neighborhood is assumed around each agent of the combined swarm, which first randomly generates a new position in this neighborhood using (1.55) and (1.56). Then, if the value of the objective function improves, this new position replaces the previous position according to the work (1.57):

$$x_{i,j}^{P_3} = x_{i,j} + \left(lb_{local,j}^{t} + \left(ub_{local,j}^{t} - rand \cdot lb_{local,j}^{t}\right)\right), \tag{1.55}$$

$$Local\ \ bounds: \begin{cases} lb_{local,j}^{t} = \dfrac{lb_j}{t}; \\ ub_{local,j}^{t} = \dfrac{ub_j}{t}, \end{cases} \tag{1.56}$$

$$X_i = \begin{cases} X_i^{P_3},\ F_i^{P_3} < F_i; \\ X_i, \text{ else}, \end{cases} \tag{1.57}$$

where $X_i^{P_3}$ is the new generated position of the *i*-th agent of the combined flock; $x_{i,j}^{P_3}$ is the *j*-th size of the agent of the combined flock; $F_i^{P_3}$ is the value of the objective function; t is the iterative circuit, lb_j and ub_j are the lower and upper limits of the *j*-th variable. $lb_{local,j}^{t}$ and $ub_{local,j}^{t}$ are the local lower and local upper limits admissible for the *j*-th variable, respectively, for simulating a local search in the neighborhood of candidate solutions.

Step 13. *Checking the stop criterion*. The algorithm terminates when the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

Step 14. *Training of the knowledge bases of agents of the combined flock*. In this research, the learning method based on evolving artificial neural networks developed in the research [2] is used to train the knowledge bases of each agent of the combined swarm. The method is used to change the nature of movement of each agent of the combined flock, for more accurate analysis results in the future.

Step 15. *Determining the amount of necessary computing resources, intelligent decision-making support system*.

In order to prevent looping of calculations on Steps 1–14 of the method and to increase the efficiency of calculations, the system load is additionally determined. When the specified threshold of computational complexity is exceeded, the amount of software and hardware resources that must be additionally involved is determined using the method proposed in the work [30].

1.3 THE METHOD OF MANAGING THE RESOURCES OF INFORMATION NETWORKS

Information network resources were understood management: spatial resource; time resource; frequency resource; a reserve of technical devices of the information network. The destabilizing factors include: deliberate obstacles; cyberattacks aimed at denial of service; fire damage to information network elements.

Information network was presented from the standpoint of graph theory in the form of a tree. At the same time, the root of the tree was matched with the control subsystem of the second level (I_2, U_2) and the vertices of this tree, which are at a distance of one edge from the root, are Q the control subsystem of the first level (I_{11}, U_{11}),...,(I_{1q}, U_{1q}),...,(I_{1Q}, U_{1Q}). Each subsystem includes a control (identification) block I and a block management U. Let's consider Q subsystems of the zero level, which are located from the root of the tree at a distance of two edges. These subsystems represent interacting processes of exchange of flows of operational and service information in control systems P_1,..., P_q,..., P_Q [4–6].

For the q-th control subsystem of the first level (I_{1q}, U_{1q}) $q=\overline{1,Q}$, let's introduce the following notations:

– $X_{1q}(k)$ is a set of vectors, the state of the q-th controlled subnet, where $x_{1q}(k)=\{x_{1q}^a(k)\}$, $a=\overline{1,a_{1q}}$, dimension $a_{1q}\times 1$;

– $\tilde{X}_{1q}(k)$ is a set of evaluation vectors $\tilde{x}_{1q}(k)=\{\tilde{x}_{1q}^a(k)\}$, $a=\overline{1,a_{1q}}$, dimensions $a_{1q}\times 1$;

– $U_{1q}(k)$ is the set of control vectors of the q-th controlled subnet $u_{1q}(k)=\{u_{1q}^b(k)\}$, $b=\overline{1,b_{1q}}$, dimension $b_{1q}\times 1$;

– $Y_{1q}(k)$ is a set of vectors of local variables that are issued to the upper-level control subsystem $y_{1q}(k)=\{y_{1q}^d(k)\}$, $d=\overline{1,d_{1q}}$, dimension $d_{1q}\times 1$;

– $Z_{1q}(k)$ is a set of vectors of local output variables $z_{1q}(k)=\{z_{1q}^d(k)\}$, $d=\overline{1,d_{1q}}$, dimension $d_{1q}\times 1$.

For the second-level management subsystem, respectively: $\tilde{X}_2(k)$ is a set of vectors of generalized estimates $\tilde{x}_2(k)=\{\tilde{x}_2^l(k)\}$, $l=\overline{1,l_r}$, dimensions $l_r\times 1=\left(\sum_{q=1}^{Q}a_{1q}\right)\times 1$;

– $Y_{2q}(k)$ is a set of vectors that are issued to the lower-level control subsystem $y_{2q}(k)=$ $=\{y_{2q}^d(k)\}$, $d=\overline{1,d_{2q}}$, dimension $d_{1q}\times 1$;

– $Z_{2q}(k)$ is a set of vectors coordinating the output variables issued to the lower-level control subsystems $z_{2q}(k)=\{z_{2q}^d(k)\}$, $d=\overline{1,d_{2q}}$, dimensions $d_{2q}\times 1$.

As a result, for the q-th subsystem of zero level P_q, $q=\overline{1,Q}$, there are: $c_{qp}(k)$ is a set of connection vectors $c_{qp}(k)=\{c_{qp}^{mn}(k)\}$, $m=\overline{1,m_q}$, $n=\overline{1,n_q}$, between p-th and q-th subsystems (p, $q=\overline{1,Q}$, $p\neq q$); $\pi_q(k)$ is a set of vectors of external influences $\Pi_q(k)=\{\pi_q^l(k)\}$, $l=\overline{1,l_q}$, dimension $l_q\times 1$.

To the set of state vectors of the information network $X(k)=\bigcup_{q=1}^{Q}X_{1q}(k)$ may include vectors of any state variables that affect the quality of the information network and the efficiency of the process of its functioning.

The main ones include:

– the vector of parameters of the information load of the information network (characterizes the number of information messages that must be transmitted per unit of time at a given bandwidth):

$$\Lambda(k)=\left\|\Lambda_1(k),\ \ldots,\ \Lambda_q(k),\ \ldots,\ \Lambda_Q(k)\right\|^T; \tag{1.58}$$

– the vector of delays in the transmission of information messages of the information network (characterized by the deterioration of the bandwidth of the information network):

$$H(k)=\left\|H_1(k),\ \ldots,\ H_q(k),\ \ldots,\ H_Q(k)\right\|^T; \tag{1.59}$$

– the vector of parameters of the radio-electronic situation of the information network (the number of suppressed operating frequencies by devices of radio-electronic warfare that do not meet the bandwidth requirements):

$$\aleph(k)=\left\|\aleph_1(k),\ \ldots,\ \aleph_q(k),\ \ldots,\ \aleph_Q(k)\right\|^T; \tag{1.60}$$

– the vector of frequency resources of the information network (total number of operating frequencies):

$$\Im(k)=\left\|\Im_1(k),\ \ldots,\ \Im_q(k),\ \ldots,\ \Im_Q(k)\right\|^T; \tag{1.61}$$

– the vector of hardware resources of the information network (total number of network devices in the information network):

$$\Upsilon(k)=\left\|\upsilon_1(k),\ \ldots,\ \upsilon_q(k),\ \ldots,\ \upsilon_Q(k)\right\|^T. \tag{1.62}$$

The development of an algorithm for complex management of information network resources.

The improved method of complex management of information network resources consists of the following sequence of steps (**Fig. 1.1**):

1. *Input of initial data.* Initial data on the state of the information network are entered.

2. *Entering information about the degree of a priori uncertainty about the state of the communication system.* At this stage, the degree of uncertainty of data on the state of the information network is determined based on the works of the authors [16–21, 28]. Possible degrees of uncertainty of information about the state of the information network: complete awareness, partial uncertainty, complete uncertainty.

3. *Determination of controlling influences on the information network.* At this stage, on the basis of the description of the state of the information network, controlling influences on the physical, channel and network levels of the information network are determined. The basis of

the specified procedure of the improved method is the method developed by the authors in previous researches [2, 3, 16, 17].

4. *Forecasting the state of the information network.* At this stage, the state of the information network with a defined composition of forces and devices of communication is forecasted. Forecasting the state of the information network at this stage is carried out using the approaches developed in previous researches [2, 3, 16, 17–21].

5. *Determination of the necessary forces and devices of communication, which are necessary for building up information network.*

The decision to increase the composition of forces and devices of communication of the information network is made after the impossibility of solving tasks related to the organization of communication in the existing organizational and staff structure after Step 4.

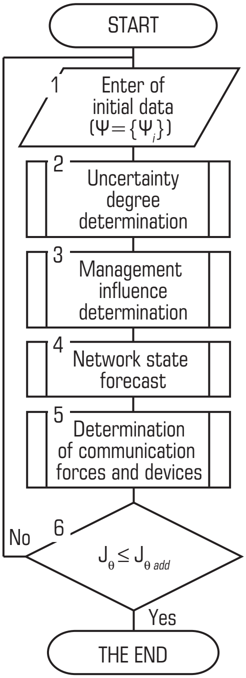

Fig. 1.1 Algorithm for the implementation of the improved method of complex management of information network resources

Let's consider in more detail the specified procedure for determining the necessary forces and devices of communication in the information network. The task of synthesizing information method

for managing the processes of coordinated functioning and complex management of information network resources can be formulated as "the task of finding optimal controlling influences". The specified tasks transfer the information network under consideration from the given to the required structural state, which characterizes both the current state of the objects included in the given type of structure and the state of relations between them.

Thus, the value of the state of the existing and "new" information network $CTC < U^t, S_\delta^{*tf} >$ is necessary [3, 7, 9, 17]:

$$J_\theta\left(X_\chi^t, \Gamma_\chi^t, \mathrm{Z}_\chi^t, F_{<\chi,\chi'>}^t, \prod\nolimits_{(\delta,\tilde{\delta})}^t, t \in (t_0, t_f]\right) \to \underset{<U^t, S_\delta^{*tf} > \in \Delta g}{extr};$$

$$\Delta_g\left\{< U^t, S_\delta^{tf} > \middle| R_\beta\left(X_\chi^t, \Gamma_\chi^t, \mathrm{Z}_\chi^t, F_{<\chi,\chi'>}^t, \prod\nolimits_{(\delta,\tilde{\delta})}^t\right) \le \tilde{R}_g\right\}; \quad (1.63)$$

$$U^t = \prod\nolimits_{<\delta_1,\delta_2>}^{t_1} \circ \prod\nolimits_{<\delta_2,\delta_3>}^{t_2} \prod\nolimits_{<\delta,\delta>}^{t_f}; \beta \in \mathbf{B}\},$$

where J_θ is the cost, time, resource indicators characterizing the quality of information network functioning; $\theta \in \Theta$ is a set of indicator numbers; χ is a set of indices corresponding to the structures of the information network; $T \in (t_0,\ t_f]$ is the time interval during which the information network functions and the communication organization process is implemented; $X_\chi^t = \{X_{\chi l}^t, l \in L_\chi\}$ is a set of elements that are part of the structure of the dynamic alternative system graph (DASG) G_χ^t (a set of DASG vertices), which is used to set the controlled structural dynamics of the information network at the moment of time t; $\Gamma_\chi^t = \{\chi_{<\chi l, l'>}^t, l, l' \in L_\chi\}$ is a set of arcs of the DASG type G_χ^t, reflect the relationships between its elements at time t; $\mathrm{Z}_\chi^t = \{z_{<\chi l, l'>}^t, l, l' \in L_\chi\}$ is a set of parameter values that quantitatively characterize the interrelationship of the corresponding elements of DASG; $F_{<\chi,\chi'>}^t$ is a description of the influence of different structures of the information network on each other at the moment of time t; $\prod\nolimits_{<\tilde{\delta},\tilde{\tilde{\delta}}>}^t$ are the compositions of the structural state of the information network with numbers $\tilde{\delta}, \tilde{\tilde{\delta}}$ at time t; Δ_g is a set of dynamic alternatives (a set of structures and parameters of the information network, "new" and existing *c* information network and a set of programs for their operation); U^t are controlling influences that allow to synthesize the structures of the information network that is being built up and implemented; $\tilde{R}_g$ are given values; "$\circ$" is an image composition operation; *B* is a set of numbers of space-time, technical and technological restrictions that determine the processes of implementation of programs for building up and functioning of a special purpose communication system [2].

At the stage of expansion, first of all, the parameters of the functioning of the elements and subsystems of the information network are changed. The research proposes to consider the information network as a complex dynamic object consisting of a set of structures. Communication between them takes place through the transmission of information about the status of operations, the intensity of data transmission and processing flows, information about the state of various resources, services and basic services.

A similar approach allows to present the stage of parallel functioning and expansion of the information network as a process of updating (improving the characteristics) of information services that support the information network. Let's present the process of software management of the expansion of the information network:

$$\frac{dx_n^{(s,l)}}{dt} = \sum_{r=1}^{p_s} u_{n\,r}^{(s,l)}(t), \tag{1.64}$$

$$\frac{dx_n^{(s,l)}}{dt} = \sum_{n=1}^{m_j} w_{n\,r}^{(s,l)}(t), \tag{1.65}$$

$$\frac{dx_{rS_l}^{(s,l)}}{dt} = \omega_{rS_l}^{(s,l)}(t). \tag{1.66}$$

The restrictions on management actions:

$$0 \le u_{n\,r}^{(s,l)}(t) \le \left[e_{n\,r}^{(s,l)}\left(1-\gamma_r^{(m,\delta)}(t)\right) + \bar{e}_{n\,r}^{(j)}\gamma_r^{(m,\delta)}(t)\right] w_{n\,r}^{(s,l)}, \tag{1.67}$$

$$\sum_{l=1}^{k_v}\sum_{n=1}^{m_j} V_n^{(s,l)} w_{n\,r}^{(s,l)}(t) \le \left[V_r^{(j)}\left(1-\gamma_r^{(m,\delta)}(t)\right) + \bar{V}_r^{(j)}\gamma_r^{(m,\delta)}(t)\right], \tag{1.68}$$

$$\sum_{l=1}^{k_v}\sum_{n=1}^{m_j} u_n^{(s,l)}(t) \le \left[P_r^{(j)}\left(1-\gamma_r^{(m,\delta)}(t)\right) + \bar{P}_r^{(j)}\gamma_r^{(m,\delta)}(t)\right], \tag{1.69}$$

$$\sum_{r=1}^{p_s} w_{n\,r}^{(s,l)}(t)\left[\sum_{\pi\in G_c}\left(\alpha_\pi^{(s,l)} - x_\pi^{(s,l)}\right) + \sum_{k\in G_c}\left(\alpha_k^{(m,r)} - x_k^{(m,r)}\right)\right] = 0, \forall l, \tag{1.70}$$

$$\sum_{r=1}^{p_s} w_{n\,r}^{(s,l)}(t) \le \varepsilon_n, \forall n;\ \sum_{n=1}^{m_j} w_{n\,r}^{(s,l)}(t) \le \theta_r, \forall r, \tag{1.71}$$

$$\omega_{rS_l}^{(s,l)}\left(a_{S_l}^{(s,l)} - x_{S_l}^{(s,l)}\right) = 0, \tag{1.72}$$

$$w_{nr}^{(s,l)} \in \left\{0, u_{vln}^{(b,j)}\right\}; \gamma_r^{(m,\delta)}(t), \omega_{rS_l}^{(s,l)} \in \{0,1\}. \tag{1.73}$$

Boundary conditions:

$$\begin{gathered} \text{for } t = t_0: x_n^{(s,b)}(t_0) = x_r^{(s,l)}(t_0) = x_{rS_l}^{(s,l)}(t_0) = 0, \\ \text{for } t = t_f: x_n^{(s,l)}(t_f) = a_n^{(s,l)};\ x_r^{(s,l)}(t_f),\ x_{rS_l}^{(s,l)}(t_f) = x_{rS_l}^{(s,l)}(t_0) \in \mathbf{R}^1. \end{gathered} \tag{1.74}$$

The indicators of the quality of software management of the expansion of the special purpose communication system:

$$J_4 = \sum_{n=1}^{m_j} \sum_{r=1}^{p_s} \int_{t_0}^{t_f} \delta_{nr}^{(s,l)}(\tau) \cdot w_{nr}^{(s,l)}(\tau) d\tau, \quad (1.75)$$

$$J_5 = \sum_{l=1}^{k_v} \sum_{r=1}^{p_s} \sum_{r=1}^{m_j} \int_{t_0}^{t_f} c_{nr}^{(s,l)}(\tau)\, w_{nr}^{(s,l)}(\tau) d\tau, \quad (1.76)$$

$$J_6 = \frac{1}{2} \sum_{l=1}^{k_v} \sum_{n=1}^{m_j} \left(\alpha_n^{(s,l)} - x_n^{(s,l)}(t_f)\right)^2, \quad (1.77)$$

where $x_r^{(s,l)}$ is the variable that characterizes the state of execution of the operation of providing the necessary information services for the performance of communication tasks $A_v^{(b,j)}$. The upper index "*s*" means that the corresponding variable is included in the model of program management of the expansion of the information network. The upper index "*l*" means the service information operation of the information network, which "consumes" the information service (service); $u_{nr}^{(s,l)}(t)$ is the intensity of support, $F_{<n,r>}^{(s,l)}$ service operations (internal service) as an information network resource $B_r^{(s,l)}$; $x_r^{(s,l)}$ is the variable, the current value of which is numerically equal to the total duration of the involvement of information network resources $B_r^{(s,l)}$; $w_{nr}^{(s,l)}(t)$ is the duration of resource $B_r^{(s,l)}$ use of the information network to support the information service (internal services or services) $D_{<l,n>}^{(s,b)}$ $w_{nr}^{(s,l)}(t) = 1$, if the information network resource is allocated and functioning; $x_{rS_l}^{(s,l)}$ numerically determines the time interval from the end of the information network $B_r^{(s,l)}$ of service internal $F_{<n,r>}^{(s,l)}$ to the given final moment of time; $\omega_{rSl}^{(s,l)}(t)$ is an auxiliary control action. Takes the value "1", if the information network has completed internal service maintenance $F_{<n,r>}^{(s,l)}$; $V_n^{(s,l)}$ is the amount of memory required to store initial and intermediate data, which is allocated to perform an internal service operation; $e_r^{(j)}$, $V_r^{(j)}$, $P_r^{(j)}$ is the set values (constants) characterizing the maximum intensity of implementation of internal services on the resource of the information network $B_r^{(s,l)}$ (before build-up); $\gamma_r^{(m,\delta)}(t)$ is an auxiliary control action that takes the value "1" at the moment of time t, if a transition from existing $\left(e_r^{(j)}, V_r^{(j)}, P_r^{(j)}\right)$ to new ones parameters of information resources B_j in the information network subsystem; $a_n^{(s,l)}$ is the volume of internal service operations to support a given external service; $\delta_{nr}^{(s,l)}(t)$ is the function, which allows to evaluate the overall quality of the provision internal services $F_{<n,r>}^{(s,l)}$ of the information network at the stage of joint functioning and expansion; $c_{nr}^{(s,l)}(t)$ is a cost function of time describing indirect, so operational costs (administration, technical support, etc.) associated with the operation and development of a specific information service.

Research has been conducted effectiveness of the proposed improved method of integrated resource management information network. The evaluation of the effectiveness of the management of spatial, temporal and frequency resources of the special purpose communication system was carried out (**Fig. 1.2**).

Analysis of the obtained dependencies (**Fig. 1.2**) shows that the value $f(\tilde{\tau}_C, \tilde{\tau}_\Pi)$ is less than the threshold value. Therefore, in this case, the decision is made mainly on the basis of the analysis of changes in the duration of the message, which was affected by interference ($\tilde{\tau}_C$) and pauses after it ($\tilde{\tau}_\Pi$). The results of the conducted modeling on build-up information network are given in **Table 1.2**.

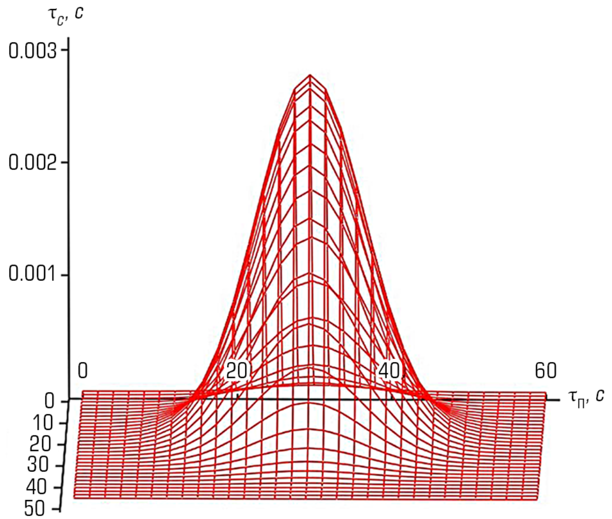

Fig. 1.2 Dependencies of the duration of messages on the reaction time of the RES complex

Table 1.2 The results of the conducted modeling on build-up information network

Output data	Option I	Option II	Option III
The number of resources that need to be increased in the system	40	50	70
The planned number of system resources to be scaled up	10	30	65
Time period during heuristic build-up	22	33	44
Time period for scaling up using the proposed approach	20	28	41
The value of the resulting quality indicator for heuristic plans	1117	901	408
The value of the resulting quality indicator for optimal plans	1384	1113	484

The results of the simulation show that increasing the efficiency of the information network under the influence of destabilizing factors by 20–26 %.

The discussion of the results of the development of the method comprehensive management of information network resources. The constraints (1.66)–(1.68) determine the possibilities of information processing on the resource $B_r^{(s,l)}$ before and after the relevant stage of building up the information network.

The constraints (1.69) determine the technology of information service implementation. The operation included in the service provision process (internal service) cannot be provided until the expansion of the information network, which is involved in this process, is completed.

The constraints (1.70) specify the possibility of simultaneous use of several information network resources to ensure the functioning of the internal service and the use of information network resources for the parallel execution of several tasks. $\in$ and θ are known numbers.

The constraints (1.71) are auxiliary and introduced in order to fix the moment of the end of the information network resources that ensure the execution of internal service operations.

The constraints (1.72) define the range of possible values that can take the corresponding control influences and forms a connection with the model of communication tasks.

The expressions (1.73) and (1.74) determine the limits on the values of the variables at given moments of time $t=t_0$ and $t=t_f$.

The type indicator (1.75) evaluates the quality of planning processes of functioning and building up the information network at the level of services.

The type indicator (1.76) is introduced to evaluate the total operating costs associated with the processes of expanding the information network only at the level of information services.

Indicator (1.77) is introduced to maximize the volume of performed operations only at the level of functioning of the special purpose communication system.

The developed and improved method allows to obtain a profit of 20–26 % compared to classical approaches to management.

CONCLUSIONS

1. In the course of the research, a mathematical model of the information conflict of the information network was proposed.

The advantages of the specified model are due to take into account a greater number of destabilizing factors, compared to the known ones. The model takes into account in the complex deliberate interference of an additive and multiplicative nature, destabilizing factors due to the presence of cyber attacks.

The disadvantages of the proposed mathematical model should be considered greater computational complexity compared to simpler mathematical models. The information conflict model of an information network can be used to develop new strategies for managing information networks, taking into account the countermeasures against the complex influence of devices of radio-electronic suppression and devices of information-technical influence. Also, the model can be used to justify new types of influences that realize the hidden functional suppression of the information network due to the creation and development of intra-system contradictions between its individual protocols.

2. The research proposed a method of self-organization of special purpose information networks, which, thanks to additional and improved procedures, allows: to take into account the type

of uncertainty and noisy data; to implement adaptive strategies for finding food sources; to combine individual swarm search strategies into a single strategy; to take into account the presence of a predator while choosing food sources; to take into account the available computing resources of the system while implementing self-organization of information networks; to change the search area by individual agents of the combined algorithm swarm; to change the speed of movement of agents of the combined algorithm swarm; to take into account the priority of searching for swarm agents of the combined algorithm; to carry out the initial display of individuals of the flock of the combined algorithm, to take into account the type of uncertainty; to carry out accurate training of individuals of the flock of the combined algorithm; to determine the best individuals of the flock of the combined algorithm with the help of an improved genetic algorithm; to conduct a local and global search taking into account the degree of data noise in the process of self-organization of information networks; to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole; to be used as a universal tool for solving the task of self-organization of information networks due to the hierarchical description of their self-organization process; to check the adequacy of the obtained results; to avoid the local extremum problem; to configure local and global search procedures, which allows the adaptation of this method to different information networks.

3. Simulation of the work of the specified method was carried out on the example of self-organization of the information network of the operational group of troops (forces). The specified example showed an increase in the efficiency of data processing at the level of 12–17 % due to the use of additional improved procedures of adding correction coefficients for uncertainty and noise of data, selection of combined swarm agents, crossing of different types of swarm optimization approaches and training of combined swarm agents.

4. The formalization of the task of managing the resources of the information network according to the maximum bandwidth was carried out. The specified formalized description allows to describe the processes that occur during the functioning of the information network and to determine the measures aimed at increasing the efficiency of the information network.

5. An improved method of comprehensive management of information network resources is developed.

The specified method allows: to determine the influence of destabilizing factors on the information network, to describe the information network of different architecture. Also, the mentioned method allows to determine the rational routes of information transfer and the mode of operation of communication devices. Another element of novelty of the proposed method is that it takes into account the degree of uncertainty about the state of the information network. The next element of novelty of the proposed method is forecasting the state of the information network based on vague temporal models and determining the number of necessary forces and devices of communication that must be increased for the full functioning of the special communication system.

6. The positive effect of the implementation of the developed improved method is provided by the following interrelated factors: joint use of the analysis and forecasting procedure; determination of the necessary number of forces and devices of communication, which must be introduced into the information network.

The specified method allows to increase the efficiency of the special communication system under the influence of destabilizing factors by 20–26 %, which is confirmed by the simulation results.

REFERENCES

1. Shyshatskyi, A. V., Bashkyrov, O. M., Kostyna, O. M. (2015). Rozvytok intehrovanykh system zviazku ta peredachi danykh dlia potreb Zbroinykh Syl. Ozbroiennia ta viiskova tekhnika, 1 (5), 35–40.
2. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskyi, R., Repilo, I. et al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 37–47. https://doi.org/10.15587/1729-4061.2020.203301
3. Sova, O., Shyshatskyi, A., Salnikova, O., Zhuk, O., Trotsko, O., Hrokholskyi, Y. (2021). Development of a method for assessment and forecasting of the radio electronic environment. EUREKA: Physics and Engineering, 4, 30–40. https://doi.org/10.21303/2461-4262.2021.001940
4. Pievtsov, H., Turinskyi, O., Zhyvotovskyi, R., Sova, O., Zvieriev, O., Lanetskii, B., Shyshatskyi, A. (2020). Development of an advanced method of finding solutions for neuro-fuzzy expert systems of analysis of the radioelectronic situation. EUREKA: Physics and Engineering, 4, 78–89. https://doi.org/10.21303/2461-4262.2020.001353
5. Zuiev, P., Zhyvotovskyi, R., Zvieriev, O., Hatsenko, S., Kuprii, V., Nakonechnyi, O. et al. (2020). Development of complex methodology of processing heterogeneous data in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 14–23. https://doi.org/10.15587/1729-4061.2020.208554
6. Shyshatskyi, A., Zvieriev, O., Salnikova, O., Demchenko, Ye., Trotsko, O., Neroznak, Ye. (2020). Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering, 9 (4), 5583–5590. https://doi.org/10.30534/ijatcse/2020/206942020
7. Nechyporuk, O., Sova, O., Shyshatskyi, A., Kravchenko, S., Nalapko, O., Shknai, O. et al. (2023). Development of a method of complex analysis and multidimensional forecasting of the state of intelligence objects. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 31–41. https://doi.org/10.15587/1729-4061.2023.276168
8. Koval, V., Nechyporuk, O., Shyshatskyi, A., Nalapko, O., Shknai, O., Zhyvylo, Y. et al. (2023). Improvement of the optimization method based on the cat pack algorithm. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 41–48. https://doi.org/10.15587/1729-4061.2023.273786

9. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. https://doi.org/10.15587/1729-4061.2019.180197
10. Mahdi, Q. A., Shyshatskyi, A., Prokopenko, Y., Ivakhnenko, T., Kupriyenko, D., Golian, V. et al. (2021). Development of estimation and forecasting method in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 51–62. https://doi.org/10.15587/1729-4061.2021.232718
11. Kuchuk, N., Merlak, V., Skorodelov, V. (2020). A method of reducing access time to poorly structured data. Advanced Information Systems, 4 (1), 97–102. https://doi.org/10.20998/2522-9052.2020.1.14
12. Tarkhan, A. B., Zhuravskyi, Y., Shyshatskyi, A., Pluhina, T., Dudnyk, V., Kiris, I. et al. (2023). Development of a solution search method using an improved fish school algorithm. Eastern-European Journal of Enterprise Technologies, 4 (4 (124)), 27–33. https://doi.org/10.15587/1729-4061.2023.284315
13. Koval, V., Shyshatskyi, A., Ransevych, R., Gura, V., Nalapko, O., Shypilova, L. et al. (2023). Development of a method for the search of solutions in the sphere of national security using bio-inspired algorithms. Eastern-European Journal of Enterprise Technologies, 3 (4 (123)), 6–13. https://doi.org/10.15587/1729-4061.2023.280355
14. Koval, M., Sova, O., Orlov, O., Shyshatskyi, A., Artabaiev, Y., Shknai, O. et al. (2022). Improvement of complex resource management of special-purpose communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 34–44. https://doi.org/10.15587/1729-4061.2022.266009
15. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The Method of Improving the Efficiency of Routes Selection in Networks of Connection with the Possibility of Self-Organization. International Journal of Advanced Trends in Computer Science and Engineering, 8 (1.2), 1–6. https://doi.org/10.30534/ijatcse/2019/0181.22019
16. Nalapko, O., Shyshatskyi, A., Ostapchuk, V., Mahdi, Q. A., Zhyvotovskyi, R., Petruk, S. et al. (2021). Development of a method of adaptive control of military radio network parameters. Eastern-European Journal of Enterprise Technologies, 1 (9 (109)), 18–32. https://doi.org/10.15587/1729-4061.2021.225331
17. Alieinykov, I., Thamer, K. A., Zhuravskyi, Y., Sova, O., Smirnova, N., Zhyvotovskyi, R. et al. (2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. Eastern-European Journal of Enterprise Technologies, 6 (2 (102), 16–27. https://doi.org/10.15587/1729-4061.2019.184394
18. Nalapko, O., Sova, O., Shyshatskyi, A., Protas, N., Kravchenko, S., Solomakha, A. et al. (2021). Analysis of methods for increasing the efficiency of dynamic routing protocols in telecommunication networks with the possibility of self-organization. Technology Audit

and Production Reserves, 5 (2 (61), 44–48. https://doi.org/10.15587/2706-5448.2021.239096

19. Sova, O., Radzivilov, H., Shyshatskyi, A., Shevchenko, D., Molodetskyi, B., Stryhun, V. et al. (2022). Development of the method of increasing the efficiency of information transfer in the special purpose networks. Eastern-European Journal of Enterprise Technologies, 3 (4 (117), 6–14. https://doi.org/10.15587/1729-4061.2022.259727
20. Sova, O., Zhuravskyi, Y., Vakulenko, Y., Shyshatskyi, A., Salnikova, O., Nalapko, O. (2022). Development of methodological principles of routing in networks of special communication in conditions of fire storm and radio-electronic suppression. EUREKA: Physics and Engineering, 3, 159–166. https://doi.org/10.21303/2461-4262.2022.002434
21. Sova, O., Radzivilov, H., Shyshatskyi, A., Shvets, P., Tkachenko, V., Nevhad, S. et al. (2022). Development of a method to improve the reliability of assessing the condition of the monitoring object in special-purpose information systems. Eastern-European Journal of Enterprise Technologies, 2 (3 (116)), 6–14. https://doi.org/10.15587/1729-4061.2022.254122
22. Shyshatskyi, A. V., Sova, O. Ya., Zhuravskyi, Yu. V., Trotsko, O. O. (2022). Metodolohichni zasady intelektualnoi obrobky danykh v intelektualnykh systemakh pidtrymky pryiniattia rishen. Theoretical and scientific foundations in research in Engineering. Boston: Primedia eLaunch, 241–269. https://doi.org/10.46299/isg.2022.mono.tech.1.3.3
23. Romanov, O. M., Shyshatskyi, A. V., Nalapko, O. L. (2022). Rozrobka metodu pidvyshchennia operatyvnosti peredachi informatsii v merezhakh spetsialnoho pryznachennia. Moderní aspekty vědy: XXI. Mezinárodní Ekonomický Institut s.r.o., 381–403. Available at: http://perspectives.pp.ua/public/site/mono/monography-21.pdf
24. Mohammed, B. A., Zhuk, O., Vozniak, R., Borysov, I., Petrozhalko, V., Davydov, I. et al. (2023). Improvement of the solution search method based on the cuckoo algorithm. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 23–30. https://doi.org/10.15587/1729-4061.2023.277608
25. Mamoori, G. A., Sova, O., Zhuk, O., Repilo, I., Melnyk, B., Sus, S. et al. (2023). The development of solution search method using improved jumping frog algorithm. Eastern-European Journal of Enterprise Technologies, 4 (3 (124)), 45–53. https://doi.org/10.15587/1729-4061.2023.285292
26. Shyshatskyi, A., Romanov, O., Shknai, O., Babenko, V., Koshlan, O., Pluhina, T. et al. (2023). Development of a solution search method using the improved emperor penguin algorithm. Eastern-European Journal of Enterprise Technologies, 6 (4 (126)), 6–13. https://doi.org/10.15587/1729-4061.2023.291008
27. Thamer, K. A., Sova, O., Shaposhnikova, O., Yashchenok, V., Stanovska, I., Shostak, S. et al. (2024). Development of a solution search method using a combined bio-inspired algorithm. Eastern-European Journal of Enterprise Technologies, 1 (4 (127)), 6–13. https://doi.org/10.15587/1729-4061.2024.298205

28. Shyshatskyi, A. V., Zhuk, O. V., Neronov, S.M., Protas, N. M., Kashkevych S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam metaevrystychnykh alhorytmiv. C91 Moderní aspekty vědy: XL. Mezinárodní Ekonomický Institut s.r.o., 529–557. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
29. Shyshatskyi, A. V., Matsyi, O. B., Yashchenok, V. Zh., Trotsko, O. O., Kashkevych, S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam kombinovanykh metaevrystychnykh alhorytmiv. C91 Moderní aspekty vědy: XL. Mezinárodní Ekonomický Institut s.r.o., 558–594. Available at: URL: http://perspectives.pp.ua/public/site/mono/mono-40.pdf

DOI: 10.15587/978-617-8360-04-7.CH2

Svitlana Kashkevich, Andrii Shyshatskyi, Oksana Dmytriieva,
Yevhen Zhyvylo, Ganna Plekhova, Serhii Neronov

CHAPTER 2

THE DEVELOPMENT OF MANAGEMENT METHODS BASED ON BIO-INSPIRED ALGORITHMS

ABSTRACT

In this chapter of the research, management methods based on bio-inspired algorithms are proposed. The basis of the research is the theory of artificial intelligence. The method is aimed at solving optimization tasks, variable solutions are defined in such a way that complex dynamic objects work in their best point (mode) according to the optimization criteria determined. In the research, the authors proposed:

– management method using a bio-inspired algorithm;

– method of finding solutions using the improved flying squirrel algorithm;

– method of assessing the state of dynamic objects using the population algorithm.

Each of the methods was based on canonical optimization algorithms, but they were improved by the authors of the research.

The essence of the improvement of these methods, which is the scientific novelty of each of them:

– the initial position of the agents is determined taking into account the type of uncertainty due to the use of a correction factor for the degree of awareness of the state of the initial situation in relation to the object of analysis;

– the initial speed of each agent is taken into account, which makes it possible to research complex functions;

– the speed of movement is regulated according to the priority of finding a solution;

– using the procedure of deep learning of knowledge bases of algorithm agents due to learning both architecture and parameters of artificial neural networks;

– select the best individuals in the flock by using an improved genetic algorithm, which improves the reliability of the decisions made.

A limitation of the research is the need to have an initial condition database complex dynamic object, the need to take into account the time delay for collecting and proving information from sources of information extraction.

It is advisable to use the proposed approach to solve the tasks of evaluating complex and dynamic processes characterized by a high degree of complexity.

KEYWORDS

Optimization problems, dynamic objects, multiagent systems, genetic algorithms, reliability and adequacy.

2.1 THE DEVELOPMENT OF A MANAGEMENT METHOD USING A BIO-INSPIRED ALGORITHM

Optimization is a complex process of identifying multiple solutions for a variety of functions. Many of today's management solutions are based on the successful solution of optimization tasks [1–3]. While solving optimization tasks, solution variables are defined in such a way that complex dynamic objects work in their best point (mode) according to the optimization criterion determined.

In essence, optimization problems of complex dynamic objects are discontinuous, undifferentiated and multimodal. Thus, it is impractical to use classic gradient deterministic algorithms [4–6] to solve optimization problems of complex dynamic objects.

To overcome the shortcomings of classical optimization algorithms for solving optimization problems of complex dynamic objects, a significant number of stochastic optimization algorithms, known as bio-inspired algorithms, were created [7–10].

One of the types of algorithms for stochastic optimization of complex dynamic objects are algorithms of swarm intelligence (swarm algorithms). Swarm intelligence algorithms are based on swarm movement and simulate interactions between the swarm and its environment to improve knowledge of the environment, such as new food sources. The most famous swarm algorithms are the particle swarm optimization algorithm, the artificial bee colony algorithm, the ant colony optimization algorithm, the wolf flock optimization algorithm and the sparrow flock algorithm [11–18].

Unfortunately, most of the basic bio-inspired algorithms mentioned above are unable to balance research and usage, resulting in unsatisfactory performance for real-world complex optimization tasks.

This prompts the implementation of various strategies to improve the convergence speed and accuracy of basic bio-inspired algorithms.

Considering the above, an urgent scientific task is the development of a management method using a bio-inspired algorithm, which would make it possible to increase the efficiency of the decisions made regarding the management of the parameters of complex dynamic objects with a given reliability.

An analysis of the state of development of the scientific direction. The work [9] presented the cognitive modeling algorithm. The main advantages of cognitive tools are determined. The lack of consideration of the type of uncertainty about the state of the object of analysis should be attributed to the shortcomings of this approach.

The work [10] revealed the essence of cognitive modeling and scenario planning. A system of complementary principles of building and implementing scenarios is proposed, different approaches to building scenarios are highlighted, the procedure for modeling scenarios based on fuzzy cognitive

maps is described. The approach proposed by the authors does not allow to take into account the type of uncertainty about the state of the analysis object and does not take into account the noise of the initial data.

The work [11] carried out an analysis of the main approaches to cognitive modeling. Cognitive analysis allows: to investigate problems with unclear factors and relationships; to take into account changes in the external environment and use objectively formed trends in the development of the situation in one's interests. At the same time, the issue of describing complex and dynamic processes remains unexplored in the work.

The work [12] presents a method of analyzing large data sets. The specified method is focused on finding hidden information in large data sets. The method includes the operations of generating analytical baselines, reducing variables, detecting sparse features and specifying rules. The disadvantages of this method include impossibility of taking into account different decision evaluation strategies, not taking into account the type of uncertainty of the input data.

The work [13] presents the mechanism of transformation of information models of construction objects to their equivalent structural models. This mechanism is designed to automate the necessary conversion, modification and addition operations during such information exchange. The shortcomings of the mentioned approach include the impossibility of assessing the adequacy and reliability of the information transformation process and the appropriate correction of the obtained models.

The work [14] developed an analytical web-platform for the research of geographical and temporal distribution of incidents. Web-platform contains several information panels with statistically significant results by territory. The disadvantages of the specified analytical platform include the impossibility of assessing the adequacy and reliability of the information transformation process, and high computational complexity. Also, one of the shortcomings of the mentioned research should be attributed to the fact that the search for a solution is not unidirectional.

The work [15] developed a method of fuzzy hierarchical assessment of library service quality. The specified method allows to evaluate the quality of libraries based on a set of input parameters. The disadvantages of the specified method include the impossibility of assessing the adequacy and reliability of the assessment and, accordingly, determining the assessment error.

The work [16] carried out an analysis of 30 algorithms for processing large data sets. Their advantages and disadvantages are shown. It has been established that the analysis of large data sets should be carried out in layers, take place in real time and have the opportunity for self-learning. The disadvantages of these methods include their high computational complexity and the impossibility of checking the adequacy of the obtained estimates.

The work [17] presents an approach for evaluating input data for decision making support systems. The essence of the proposed approach consists in the clustering of the basic set of input data, their analysis, after which the system is trained based on the analysis. The disadvantages of this approach are the gradual accumulation of assessment and training errors due to the lack of an opportunity to assess the adequacy of the decisions made.

The work [18] presents an approach to data processing from various sources of information. This approach allows to process data from various sources. The disadvantages of this approach include the low accuracy of the obtained estimate and the impossibility of verifying the reliability of the obtained estimate.

The work [19] carried out a comparative analysis of existing decision-making support technologies, namely: the method of analyzing hierarchies, neural networks, the theory of fuzzy sets, genetic algorithms and neuro-fuzzy modeling. The advantages and disadvantages of these approaches are indicated. The spheres of their application are defined. It is shown that the method of analyzing hierarchies works well under the condition of complete initial information, but due to the need for experts to compare alternatives and select evaluation criteria, it has a high share of subjectivity. For forecasting problems under conditions of risk and uncertainty, the use of the theory of fuzzy sets and neural networks is justified.

The work [20] developed a method of structural and objective analysis of the development of weakly structured systems. An approach to the research of conflict situations caused by contradictions in the interests of subjects that affect the development of the studied system and methods of solving poorly structured problems based on the formation of scenarios for the development of the situation. At the same time, the problem is defined as a discrepancy between the existing state of the system and the required one set by the management entity. At the same time, the disadvantages of the proposed method include the problem of the local optimum and the inability to conduct a parallel search.

The work [21] presents a cognitive approach to simulation modeling of complex systems. The advantages of the specified approach, which allows to describe the hierarchical components of the system, are shown. The shortcomings of the proposed approach include the lack of consideration of the computing resources of the system.

The work [22] indicated that the most popular evolutionary bio-inspired algorithms are the so-called "swarm" procedures (Particle Swarm Optimization – PSO). Among them, there are optimization algorithms based on cat swarms (Cat Swarm Optimization – CSO), which are very promising both from the point of view of speed and ease of implementation. These algorithms have proven their effectiveness in solving a number of rather complex tasks and have already undergone a number of modifications. Among the modifications, procedures based on harmonic search, fractional derivatives, adaptation of search parameters and, finally, "crazy cats" can be noted. At the same time, these procedures are not without some shortcomings that worsen the properties of the global extremum search process.

An analysis of works [9–22] showed that the common shortcomings of the above-mentioned researches are:

– the lack of possibility of forming a hierarchical system of indicators, according to which the state of complex dynamic objects is assessed;

– the lack of consideration of computing resources of the system that evaluates the state of complex dynamic objects;

– the lack of mechanisms for adjusting the system of indicators for assessing the state of complex dynamic objects;

– a failure to take into account the type of uncertainty and noise of data on the state of complex dynamic objects, which creates corresponding errors while assessing their real state;

– the lack of deep learning mechanisms of knowledge bases;

– high computational complexity;

– the lack of consideration of computing (hardware) resources available in the system;

– the lack of search priority in a certain direction.

The aim of the research is the development of a management method using a bio-inspired algorithm. This will allow to increase the speed of assessment of the state of dynamic objects with a given reliability and the development of subsequent management decisions. This will make it possible to develop software for intelligent decision-making support systems.

To achieve the aim, the following tasks were set:

– to determine the method implementation algorithm;

– to lead an example of the application of the method in the analysis of the operational situation of a group of troops (forces).

The problem, which is solved in the research, is to increase the efficiency of decision making in the tasks of assessing the state of dynamic objects while ensuring the given reliability regardless of its hierarchy. *The object of research* is complex dynamic objects with a hierarchical construction structure. *The subject of research* is the process of decision making in management tasks with the help of an improved goose flock algorithm (GFA), an improved genetic algorithm and evolving artificial neural networks.

Research hypothesis is a possibility of increasing the efficiency of decision-making with a given assessment reliability using a combined swarm algorithm.

The proposed method was simulated in the MathCad 14 software environment (USA). The assessment of the elements of the operational situation of the group of troops (forces) was used as a task to be solved during the simulation. The hardware of the research process is AMD Ryzen 5.

The operational grouping of troops (forces) was considered as an object of assessment. An operational grouping of troops (forces) formed on the basis of an operational command with a typical composition of forces and devices according to the wartime staff and with a range of responsibility in accordance with current regulations.

The research is based on the goose flock algorithm – to find a solution regarding the state of dynamic objects with a hierarchical structure. To train the individuals of the combined swarm algorithm, evolving artificial neural networks are used and to select the best ones in the combined swarm algorithm, an improved genetic algorithm is used.

The proposed approach is a bio-inspired algorithm that assumes that AG form a swarm. This approach is able to provide appropriate solutions for optimization problems (for subsequent management) in a multiple iterative process based on the ability to search for its members (AG) in the problem-solving space.

Each member of the AG flock, based on its position in space, determines the values for the variables of the problem solution. Thus, each AG, as a member of the population, is a candidate for solving the problem, which is modeled from a mathematical point of view using a vector.

The management method using a bio-inspired algorithm consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, it is determined:

– the number of AG in the flock;

– the maximum value of the fitness function;

– the number of alpha individuals (leaders), a function that describes the object of management;

– basic positions of each AG in the search space;

– the number of iterations of the search algorithm;

– pebble weight (AG used by guards).

Step 2. Creation of AG flock. Initialization of the AG population X_i $(i=1,2,...,n)$ takes place. A set of AG form a population described by the matrix X. The location of each AG describes the i-th element of the general matrix X. AG are displayed on the search plane taking into account the uncertainty about the object of management and the basic model of its state is initialized [2, 19, 21].

The position of the AG in the task space is initialized at the beginning of the algorithm start using equation (2.1):

$$x_{i,d} = lb_d + r\cdot\left(ub_d - lb_d\right), \tag{2.1}$$

where X is the population matrix of AG; X_i is the i-th member of the AG flock; $x_{i,d}$ is the d-th dimension in the solution search space, regarding the state of the control object; N is the number of AG; m is the number of decision variables; r is the random number in the interval [0, 1]; lb_d and ub_d are the lower and upper bounds of the d solution variables.

Since the position of each AG in the task solution space represents a variant of the problem, the value of the objective function can be estimated according to the position of each AG.

Step 3. Numbering of AG in the flock, i, $i\in[0, S]$. At this stage, each AG is assigned a serial number. This makes it possible to determine the parameters of finding a solution for each individual in the flock.

Step 4. Division of AG by functional purpose. At this stage, AG is divided according to purpose. In this case, the division of available AG is carried out into two categories: AG-guards of the flock (up to 10 % of the number of the flock), the remaining AG are those that obtain food.

Step 5. Determining the initial speed of blood pressure.

Initial speed v_0 of each AG is determined by the following expression:

$$v_i = \left(v_1, v_2 ... v_S\right),\ v_i = v_0. \tag{2.2}$$

In the planning of the proposed approach, the position of the AG flock members in the problem-solving space is updated based on the simulation of the foraging strategy of the AG in the wild.

According to this, in each iteration, the position of the population members is updated in two stages: exploration based on the simulation of the movement of AG towards the food source and exploitation based on the simulation of the feeding behavior of AG. The update of the position of AG-guards is determined according to the position of the main flock.

Step 6. Preliminary evaluation of the search area of the AG. In this procedure, the search area in natural language is determined precisely by the halo of the existence of the AG. Considering that food sources for AG are plant-based food, it is advisable to sort food sources for suitability (Step 7).

Step 7. Classification of food sources for hypertension.

The location of the best food source (minimum fitness) is considered to be (FS_{ht}) the food plant (watercress) that is nearby and requires the least amount of energy to find and obtain it. Delicacy food of plant origin will be denoted as FS_{at}.

Other non-priority sources of food (food that is necessary for the survival of individuals) will be designated as FS_{nt}:

$$FS_{ht}=FS(\text{sorte\_index}(1)), \tag{2.3}$$

$$FS_{at}(1:3)=\text{FS}(\text{sorte\_index}(2:4)), \tag{2.4}$$

$$FS_{nt}(1:NP\text{-}4)=\text{FS}(\text{sorte\_index}(5:\ NP)). \tag{2.5}$$

Step 8. Formation of a flock of guards and their parameters. As indicated in Step 4 of the algorithm, a certain part of the flock remains as guards. The ability to protect the flock is a necessary condition for the exploitation phase.

Step 8.1. Calculation of the weight of the pebble that AG keeps in his paws. The pebble is intended to alert the resting AG about the approach of a predator to the flock. Accordingly, the time of its fall and the propagation of sound depends on the size of the pebble. In the natural environment, the weight of a pebble stored by an individual ranges from 1 to 25 kg.

The expression (2.6) describes the procedure for finding the weight of a pebble randomly for any iteration:

$$W_{it}=rand([1,\ 25],1,1). \tag{2.6}$$

Step 8.2. Calculating the time it takes the pebble to reach the ground.

The expression (2.7) allows to calculate the time Tst_{it} required for a falling pebble to reach the ground. This is a random value between 1 and the number of measurements for each iteration in the loop:

$$Tst_{it}=rand(1,\ \text{dim}). \tag{2.7}$$

Step 8.3. Calculation of the time of impact of a stone on the ground.

The expression (2.8) allows to calculate the Tas_{it} time, when the stone hits the ground and a sound is heard, which is transmitted to each AG in the flock:

$$Tas_{it} = rand(1, \dim). \tag{2.8}$$

Step 8.4. Calculation of sound propagation time. The expression (2.9) describes the procedure for calculating the total time T_{total}, which is required for the propagation of sound and its arrival to a separate AG in the flock during the iterations of the algorithm:

$$T_{total} = \frac{\sum (Tas_{it})}{\dim}. \tag{2.9}$$

To get the average required time T_a, the total time is divided by 2:

$$T_a = \frac{T_{total}}{2}. \tag{2.10}$$

Step 9. Formation of the AG search party. Alpha AG, who are not guardians of the flock, go in search of food sources and lead the rest of the flock.

Step 10. Checking the suitability of each AG.

The relevance of each search AG is determined in each iteration using the improved genetic algorithm proposed in works [23–26] and comparing the obtained values with standardized functions. The fitness value of each AG in the search swarm (each row in the X matrix) is measured and compared with the fitness of the remaining AG (the other rows of the X matrix).

Step 11. Checking the presence of a predator near the flock. To protect and wake up the AG in this group, it is possible to calculate the falling speed of the pebble:

$$V_{FFS} = T_{ait} \cdot \frac{\sqrt[2]{W_{it}}}{9.81}. \tag{2.11}$$

In equation (2.12), to find the sound propagation distance S_{it}, it must be the sound speed V_{ss} in air, multiplied by the sound propagation time Tas_{it}. The speed of sound in air is 343.2 m per second:

$$S_{it} = V_{ss} \cdot Tas_{it}. \tag{2.12}$$

In this step let's find D_{it} the distance between the AG guard and another AG that is resting or feeding. In equation (2.13), the sound propagation distance S_{it} multiplied by 1/2 is used, because only the time for the sound to travel is needed, not the time for the sound to return:

$$D_{it} = 1/2 \cdot S_{it}. \tag{2.13}$$

To wake an individual in a flock, it is necessary to find the $BestX_{it}$ as shown in equation (2.14). This equation consists of the falling speed of the pebble V_{FFS}, added to the distance AG D_{it}, multiplied by the average value in the square of time T_a:

$$(X_{it+1}) = V_{FFS} + D_{it} \cdot T_a^2. \tag{2.14}$$

Step 12. Checking the stop criterion. The algorithm terminates when the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

Step 13. Training of AG knowledge bases.

In the research, the learning method based on evolving artificial neural networks developed in the research [2] is used to learn the knowledge bases of each AG. The method is used to change the nature of movement of each AG, for more accurate analysis results in the future.

Step 14. Determining the amount of necessary computing resources, intelligent decision-making support system.

In order to prevent looping of calculations on Steps 1–13 of this method and to increase the efficiency of calculations, the system load is additionally determined. When the specified threshold of computational complexity is exceeded, the amount of software and hardware resources that must be additionally involved is determined using the method proposed in works [27–33].

The end of algorithm.

The proposed management method using a bio-inspired algorithm. To determine the effectiveness of the proposed method, modeling of its work was carried out to solve the task of determining the composition of the operational grouping of troops (forces) and the elements of its operational construction in order to determine the expediency of regrouping troops (forces).

Initial data for determining the composition of the operational grouping of troops (forces) and elements of its operational construction using the method:

– the number of sources of information about the state of the monitoring object is 3 (radio monitoring tools, remote sensing of the earth and unmanned aerial vehicles). To simplify the modeling, the same number of each tool was taken – 4 tools each;

– the number of informational signs by which the state of the monitoring object is determined – 12. Such parameters include: ownership, type of organizational and staff formation, priority, minimum width along the front, maximum width along the front. The number of personnel, minimum depth along the flank, maximum depth along the flank, the number of samples of weapons and military equipment (WME), the number of types of WME samples and the number of communication devices), type of operational construction are also taken into account;

– the variants of organizational and personnel formations are company, battalion, brigade.

Parameters of the method:

– the number of iterations is 100;

– the number of individuals in the flock is 50 (the number of AG guardians of the flock is 5, the number of AG foragers is 45);

– the weight of a pebble for AG guards is 2 kg;

– the range of feature space is [–150,150].

The parameters of the improved genetic algorithm:

Selection – Roulette wheel (proportional).

Crossover probability=0.8.

Mutation – Gaussian probability=0.05.

Effectiveness of the method management using a bio-inspired algorithm is compared with the swarm optimization algorithms given in **Tables 2.1–2.4**. The comparison was made with unimodal and multimodal functions.

As can be seen from the **Tables 2.1–2.4** increasing the efficiency of decision making is achieved at the level of 10–12 % due to the use of additional procedures.

It can be seen that the state estimation method of dynamic objects using the combined swarm algorithm is able to converge to the true value for most unimodal functions with the fastest convergence speed and the highest accuracy, while the convergence results of the particle swarm algorithm are far from satisfactory.

Table 2.1 Efficiency of optimization algorithms at solving the task of determining the composition of the operational grouping of troops (forces) and the elements of its operational construction

The name of the algorithm	*Ts*	Optimal variables		*L*	Optimal cost
		T_h	*R*		
Algorithm for the optimization of a flock of walruses	0.7280271	0.3845792	40.312284	200	5882.8955
Particle swarm algorithm	0.7480269	0.3845797	40.312282	200	5882.9013
Flying squirrel algorithm	0.7690308	0.384581	40.312476	199.99732	5882.9077
Artificial bee colony algorithm	1.1950157	0.64038	60.549321	48.031984	7759.8234
Ant colony algorithm	0.7780271	0.3845792	40.312284	200	5882.9013
The proposed method	0.7794994	0.385819	40.386517	200	5909.3749
Algorithm of a flock of monkeys	0.911517	0.4510723	46.230782	133.83941	6270.8621
The bat swarm algorithm	0.8344267	0.4164052	43.217775	163.90679	6003.8497
Locust swarm algorithm	0.7784599	0.3858127	40.320627	199.96442	5890.2105
Genetic algorithm	1.5622593	0.4813024	47.695987	124.64823	10,807.366
Algorithm for optimization of a flock of cats	1.1300127	1.1576349	44.110061	190.7876	11,984,417
Algorithm of invasive weeds	1.55006	0.6231249	63.139483	49.78495	9998.6395
Firefly swarm algorithm	1.406417	0.7832762	58.253368	73.964478	10,920,286

Table 2.2 Comparative analysis of the effectiveness of optimization algorithms at solving the task of determining the composition of the operational grouping of troops (forces) and the elements of its operational construction

The name of the algorithm	**Average**	**Best**	**Worst**	**Standard**	**Median**	**Rank**
Algorithm for the optimization of a flock of walruses	5882.8955	5882.8955	5882.8955	1.87E-12	5882.8955	1
Particle swarm algorithm	5891.226	5882.9013	5965.0365	22.218932	5882.9017	3
Flying squirrel algorithm	6219.5386	5882.9077	7046.3206	352.35848	6047.6955	5
Artificial bee colony algorithm	12,409,586	7759.8234	19,991,769	3127.065	11,403.338	9
Ant colony algorithm	5882.9013	5882.9013	5882.9013	3.68E-06	5882.9013	2
The proposed method	6271.132	5909.3749	6948.3792	333.1584	6143.6153	6
Algorithm of a flock of monkeys	7998.6372	6270.8621	12,805,388	1681.8974	7579.6333	8
The bat swarm algorithm	6518.1019	6003.8497	7050.4059	320.31898	6572.19	7
Locust swarm algorithm	6012.3675	5890.2105	6670.9945	239.38549	5898.5494	4
Genetic algorithm	28,273,334	10,807.366	60,311.64	13,795.65	24,975,491	12
Algorithm for optimization of a flock of cats	20,643,589	11,984,417	32,105,445	6711.6675	19,830,394	10
Algorithm of invasive weeds	29,687,575	9998.6395	50,712.307	12,915,318	32,709,339	13
Firefly swarm algorithm	25,427,766	10,920,286	45,530,922	10,828,815	22,551,255	11

Table 2.3 Comparative analysis of the effectiveness of optimization algorithms while determining the composition of an operational group of troops (forces)

No. *F*	**Value**	**Algorithm of a flock of gray wolves**	**Algorithm of a flock of walruses**	**Particle swarm algorithm**	**Algorithm of a flock of monkeys**	**Algorithm of flock of hawks**	**The bat swarm algorithm**	**Algorithm of a herd of coots**	**The proposed method**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
F_1	Average	2.22E-27	7.92E-72	2.63E+00	9.73E-48	5.75E-98	6.49E-06	4.99E-07	**1.22E-22**
	Standard	5.43E-27	4.22E-71	1.15E+00	4.80E-47	2.87E-97	1.28E-05	6.71E-07	**6.6E-22**
F_2	Average	8.23E-17	4.46E-51	1.44E-49	0.00E+00	1.51E-50	3.39E-05	1.26E-05	**2.21E-14**
	Standard	6.13E-17	1.30E-50	7.53E-49	0.00E+00	5.19E-50	4.15E-05	2.54E-05	**6.57E-14**
F_3	Average	1.43E-05	4.25E+04	1.95E+02	4.04E-03	3.50E-72	9.74E+01	1.29E-01	**4.06E-26**
	Standard	3.47E-05	1.23E+04	6.70E+01	7.64E-03	1.79E-71	1.57E+02	2.52E-01	**1.98E-26**
F_4	Average	1.08E-06	5.26E+01	1.97E+00	3.02 E-02	3.14E-49	2.87E-01	5.03E-02	**2.14E-13**
	Standard	1.26E-06	2.68E+01	2.31E-01	1.77E-02	1.47E-48	3.48E-01	8.79E-02	**9.39E-13**
F_5	Average	2.70E+01	2.80E+01	1.12E+03	2.85E+01	8.34E-03	2.89E+01	2.84E+01	**4.52E+01**
	Standard	8.74E-01	5.51E-01	6.36E+02	3.06E-01	1.36E-02	1.26E-01	4.05E-01	**3.22E+01**
F_6	Average	7.66E-01	4.69E-01	2.42E+00	3.27E+00	1.02E-04	3.72E+00	2.56E+00	**1.97E-01**
	Standard	3.53E-01	2.86E-01	1.21E+00	2.36E-01	1.16E-04	4.53E-01	4.64E-01	**1.23E-01**
F_7	Average	2.19E-03	2.54E-03	1.92E+01	9.03E-05	1.28E-04	2.09E-03	6.91E-03	**5.77E-03**
	Standard	1.22E-03	2.76E-03	1.04E+01	8.86E-05	1.01E-04	2.80E-03	5.71E-03	**5.24E-03**
F_8	Average	–6.02E+03	–1.08E+04	–6.22E+03	–5.43E+03	–1.24E+04	–5.74E+03	–5.91E+03	**–7.44E+03**
	Standard	1.02E+03	1.73E+03	1.50E+03	4.29E+02	5.03E+02	7.42E+01	5.13E+02	**7.55E+02**
F_9	Average	1.90E+00	1.75E+00	1.65E+02	0.00E+00	0.00E+00	1.66E+01	1.12E+01	**1.47E-04**
	Standard	2.74E+00	7.09E+00	2.44E+01	0.00E+00	0.00E+00	2.01 E+01	1.64E+01	**9.49E-04**

Continuation of Table 2.3

1	2	3	4	5	6	7	8	9	10
F_{10}	Average	1.00E-13	4.56E-15	2.62E+00	8.88E-16	8.88E-16	2.00E+01	2.00E+01	**2.77E-09**
	Standard	1.47E-14	2.38E-15	4.84E-01	0.00E+00	0.00E+00	1.20E-03	1.20E-03	**1.42E-08**
F_{11}	Average	4.27E-03	6.89E-03	1.22E-01	1.76E-01	0.00E+00	2.57E-02	2.59E-02	**1.28E-16**
	Standard	1.06E-02	2.77E-02	3.73E-02	1.34E-01	0.00E+00	6.67 E-02	4.07E-02	**2.78E-16**
F_{12}	Average	4.27E-02	2.20E-02	6.26E-02	5.10E+06	1.02E-05	5.41E-01	2.41E-01	**3.77E-01**
	Standard	2.01E-02	1.08 E-02	5.55E-02	4.63E+07	1.58E-05	2.24E-01	1.61E-01	**8.26E-01**
F_{13}	Average	6.82E-01	5.70E-01	5.65E-01	2.83E+00	9.94E-05	2.72E+00	1.89E+00	**4.12E-01**
	Standard	2.49E-01	2.52E-01	2.77E-01	1.02E-01	1.54E-04	1.27E-01	2.51E-01	**5.26E-01**
F_{14}	Average	4.29E+00	3.48E+00	2.88E+00	9.80E+00	1.72E+00	1.32E+00	1.23E+00	**9.98E-01**
	Standard	4.20E+00	3.66E+00	2.01E+00	4.28E+00	1.97E+00	1.78E+00	5.64E-01	**6.60E- 16**
F_{15}	Average	3.14E-03	8.30E-04	8.44E-04	1.18E-02	3.74E-04	1.30F.-03	1.09E-03	**1.32E-03**
	Standard	6.88E-03	5.59E-04	1.59E-04	1.76E-02	1.71E-04	5.13E-05	3.35E-04	**3.64E-03**
F_{16}	Average	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	**–1.08E+00**
	Standard	2.62E-08	3.12E-09	4.46E-16	1.24E-07	8.85E-10	1.46E-05	3.45E-06	**3.22E-12**
F_{17}	Average	3.98E-01	3.98E-01	3.98E 01	4.14E-01	3.98E-01	5.54E-01	3.98E-01	**4.31E-01**
	Standard	3.11E-06	3.57E-05	0.00E+00	1.54E-02	2.30E-05	8.47E-01	4.95E-04	**8.85E-07**

Table 2.4 Results of the AG algorithm for test functions

Function	The dimensionality of the function	The number of AG	The number of iterations	Result	Average time, sec.
Benchmark (global optimum: 0)	2	4	100	0	0
	5	14		0	0.1
	10	28		0.005	3.11
	30	50		0.007	30.42
Rastrigin (global optimum: 0)	2	6	100	0	0
	5	30		0	18.22
	10	50		0.03	62.2
	30	50		0.92	527.6
Hryvnoka (global optimum: 0)	2	6	100	0	0
	5	16		0.002	0.16
	10	30		0.004	4.55
	30	50		0.024	89.01
Ackley (global optimum: 0)	2	6	100	0	0
	5	24		0.001	0.15
	10	42		0.012	3.15
	30	50		0.021	65.92
Bukin (global optimum: 0)	2	8	100	0	0
	5	20		0.002	1.94
	10	50		0.02	3.97

The advantages of the proposed method are due to the following:

– the initial position of the AG is carried out taking into account the type of uncertainty (Step 2), in comparison with works [9, 14, 21];

– the initial speed of each AG is taken into account (Step 4), in comparison with works [9–15];

– suitability of the place of search for the AG is determined, which reduces the time of searching for a solution (Step 5), in comparison with works [14, 16, 17];

– universality of strategies for searching for food places of AG, which allows to classify the type of data to be processed (Steps 6, 7), in comparison with works [14, 16, 17];

– there is a classification of food sources of AG, which determines the priority of finding a solution (Step 6), in comparison with works [11, 13, 17–19];

– taking into account the presence of a predator while foraging for food, which allows to avoid local optima (Step 11), in comparison with works [12, 13, 15–18];

– accelerated selection of individuals for each AG due to the use of an improved genetic algorithm (Step 10), in comparison with works [9, 12, 13–18];

– universality of solving the task of analyzing the state of dynamic objects by swarm agents of the combined swarm algorithm due to the hierarchical nature of their description (Steps 1–14), in comparison with works [9, 12, 13–18];

– possibility of simultaneously searching for a solution in different directions (Steps 1–14, **Tables 2.1–2.3**);

– adequacy of the obtained results (Steps 1–14), in comparison with works [9–23];

– ability to avoid the local extremum problem (Steps 1–14);

– possibility of in-depth learning of AG knowledge bases (Step 14), in comparison with works [9–23];

– possibility of calculating the necessary amount of computing resources, which must be involved in case of impossibility of carrying out calculations with available computing resources (Step 14), in comparison with works [9–23].

The disadvantages of the proposed method include:

– loss of informativeness while assessing the state of complex dynamic objects due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of complex dynamic objects;

– loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the condition of complex dynamic objects;

– to determine effective measures to increase the efficiency of management of complex dynamic objects;

– to increase the speed of assessment of the state of complex dynamic objects;

– to reduce the use of computing resources of decision-making support systems.

A limitation of the research is the need to have an initial condition database complex dynamic object, the need to take into account the time delay for collecting and proving information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 21–23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

2.2 METHOD OF FINDING SOLUTIONS USING AN IMPROVED FLYING SQUIRREL ALGORITHM

The proposed algorithm is an improved flying squirrel algorithm (FSA) and consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available about the object to be analyzed are entered.

The main parameters of FSA are the maximum number of iterations $Iter_{\max}$, the population size NP, the number of decision variables n, the probability of the presence of a predator P_{dp}, the scaling factor sf, the sliding constant G_c, the upper and lower bounds for the decision variable FS_U and FS_L. These parameters are set at the beginning of the FSA procedure:

$$FS_{i,j} = FS_L + rand(\ )\cdot(FS_U - FS_L),\ i = 1,\ 2,\ldots,\ NP,\ j = 1,\ 2,\ldots,n, \tag{2.15}$$

where $rand()$ is a uniformly distributed random number in the range [0, 1].

Step 2. Exhibition agents on the search plane.

At this stage, FSA is issued taking into account the type of uncertainty about the object to be analyzed and the basic model of the object state is initialized [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. The above is carried out with the help of appropriate correction coefficients, which are set at the analysis stage.

Step 3. Numbering of FSA in the flock, i, $i \in [0, S]$.

Step 4. Setting the initial fitness function.

Fitness value $f = (f_1 f_2,\ldots, f_{NP})$ of an individual, the location of the FSA is calculated by substituting the value of the decision variables into the fitness function:

$$f_i = f_i\left(FS_{i,1}, FS_{i,2},\ldots,FS_{i,n}\right),\ i = 1,2,\ldots,\ NP. \tag{2.16}$$

Step 5. Determining the quality of food in the search area for FSA.

The quality of food sources is determined by the suitability value of the FSA location sorted in ascending order:

$$\left[sorted\_f,\ sorte\_index\right] = sort(f). \tag{2.17}$$

Step 6. Classification of trees (food sources) for FSA.

After sorting the food sources in each area, the FSA classifies three types of trees: hickory (food source – hickory nuts), oak (food source – acorns) and common tree.

The location of the best food source (minimum fitness) is considered a hickory nut tree (FS_{ht}), the following three food source locations must be nut trees (FS_{at}) and the rest are considered common trees (FS_{nt}).

Step 7. Creating new places with the help of FSA sliding. At the stage of creating new places, three main scenarios are used. Let's consider each of them in detail.

Step 7.1. Flying squirrels on acorn nut trees tend to move toward hickory nut trees. New food halos can be created in the following way:

$$FS_{at}^{new} = \begin{cases} FS_{at}^{old} + d_g G_c \left(FS_{ht}^{old} - FS_{at}^{old} \right), \text{if } R_1 \geq P_{dp}, \\ \text{random location, otherwise,} \end{cases} \quad (2.18)$$

where d_g is the random sliding distance of the FSA; R_1 is a function that returns the value of a uniform distribution on the interval [0, 1]; and G_c is the sliding constant of the FSA.

Step 7.2. Some FSA that live in common trees may move to the walnut tree to meet their daily energy needs. New food halos of FSA can be created in the following way:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c \left(FS_{at}^{old} - FS_{nt}^{old} \right), \text{if } R_2 \geq P_{dp}, \\ \text{random location, otherwise,} \end{cases} \quad (2.19)$$

where R_2 is a function that returns a value from a uniform distribution on the interval [0, 1].

Step 7.3. Some FSA on common trees may switch to hickory nuts if they have already met their daily energy needs. In this scenario, a new arrangement of proteins can be generated in the following way:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c \left(FS_{ht}^{old} - FS_{nt}^{old} \right), \text{if } R_3 \geq P_{dp}, \\ \text{random location, otherwise,} \end{cases} \quad (2.20)$$

where R_3 is a function that returns a value from a uniform distribution on the interval [0, 1].

In all scenarios, the sliding distance of the FSA d_g is considered to be between 9 and 20 m [27]. However, this value is quite large and can introduce large perturbations (2.16)–(2.19) and can lead to unsatisfactory performance of the algorithm. To achieve acceptable performance of the algorithm, the scale factor (*sf*) was introduced as a divisor of d_g and its value was chosen equal to 18 [27].

Step 8. Checking the presence of a predator. At this stage of the FSA, the presence of predators is checked. If there are predators, go to Step 9. If there are no predators, go to Step 10.

Step 9. FSA actions in the presence of a predator. When FSA establish new sites, their natural behavior is affected by the presence of predators and this is controlled by the presence of the predator probability P_{dp}. At the early stage of the search, the FSA population is often far from the food source and its distribution area is large. Thus, it faces a great threat from predators. In the course of evolution, FSA locations are near the food source (optimal solution). In this case, the distribution area of the FSA population is becoming smaller and less threat from predators is expected. Thus, to improve the performance of the FSA, the adaptive P_{dp}, which changes dynamically as a function of the iteration number, is adopted in the following way:

$$P_{dp} = \left(P_{dp\max} - P_{dp\min}\right) \times \left(1 - Iter/Iter_{\max}\right)^{10} + P_{dp\min}, \tag{2.21}$$

where P_{dpmax} and P_{dpmin} are the maximum and minimum probability of the presence of a predator, respectively.

Step 10. Checking the condition of seasonal monitoring of FSA.

Food behavior of FSA largely depends on seasonal fluctuations. Therefore, the condition of seasonal monitoring is introduced into the algorithm to prevent the algorithm from falling into local optimal solutions.

First, the seasonal constant S_c and its minimum value are calculated:

$$S_c^t = \sqrt{\sum_{k=1}^{n}\left(FS_{at,k}^t - FS_{ht,k}\right)^2},\ t = 1,2,3, \tag{2.22}$$

$$S_{c\min} = \frac{10E-6}{365^{Iter/(Iter_{\max})/2.5}}. \tag{2.23}$$

Then the seasonal monitoring condition is checked. Provided $S_c^t < S_{c\min}$, winter is over and FSA, who lose the ability to explore the forest, will again randomly move their places of finding a source of food:

$$FS_{nt}^{new} = FS_L + Levy(n) \times \left(FS_U - FS_L\right), \tag{2.24}$$

where

$$Levy(x) = 0.01 \times \frac{r_a \times \sigma}{\left|r_b\right|^{1/\beta}}, \tag{2.25}$$

where r_a and r_b are two functions that return values from a uniform *Levy* distribution on the interval [0, 1]; β is a constant ($\beta=1.5$ in the research) and σ is calculated in the following way:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{((\beta-1)/2)}}\right)^{1/\beta}, \tag{2.26}$$

where $\Gamma(x)=(x-1)!$.

Step 11. Checking the stop criterion. The algorithm terminates when the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking seasonal monitoring conditions is repeated.

Step 12. Generation of FSA positions taking into account the degree of data noise.

Under the condition R_1, R_2, $R_3 < P_{dp}$, FSA continue to slide to the next potential food locations, different individuals tend to have different judgments and their sliding directions and change procedures. Sliding takes place taking into account the degree of data noise, which is distributed from 0 to 1.

In other words, the feed behavior of FSA has the characteristics of randomness and vagueness. These characteristics can be artificially described and integrated using a conventional cloud model. In the model, a conventional cloud generator model is used instead of uniformly distributed random functions to generate a new location for each flying squirrel. Thus, (2.24)–(2.26) are replaced by the following equations:

$$FS_{at}^{new} = \begin{cases} FS_{at}^{old} + d_g G_c \left(FS_{ht}^{old} - FS_{at}^{old} \right), \text{ if } R_1 \geq P_{dp}; \\ Cx\left(FS_{at}^{old}, En, He \right), \text{ otherwise,} \end{cases} \quad (2.27)$$

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c \left(FS_{at}^{old} - FS_{nt}^{old} \right), \text{ if } R_2 \geq P_{dp}; \\ Cx\left(FS_{nt}^{old}, En, He \right), \text{ otherwise,} \end{cases} \quad (2.28)$$

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c \left(FS_{ht}^{old} - FS_{nt}^{old} \right), \text{ if } R_3 \geq P_{dp}; \\ Cx\left(FS_{nt}^{old}, En, He \right), \text{ otherwise,} \end{cases} \quad (2.29)$$

where *En* (entropy) represents the uncertainty measurement of a qualitative concept and *He* (hyper-entropy) is the uncertain degree of entropy *En* [32]. In particular, in (2.27)–(2.29) *En* means the search radius, and $He = 0.1\ En$ is used to represent search stability. In the early iterations, a large *En* is required because the location of the flying proteins is often far from the optimal solution. Under the condition of finite generations, where the population location is close to the optimal solution, a smaller *En* is suitable for fine-tuning the solutions. Therefore, the search radius *En* changes dynamically with the iteration number:

$$En = En_{\max} \times \left(1 - Iter/Iter_{\max} \right)^{10} + P_{dp\min}, \quad (2.30)$$

where $En_{\max} = (FS_U - FS_L)/4$ is the maximum search radius.

Step 13. Accelerating the intensity of the search for feeding areas of FSA.

In the basic FSA, all dimensions of one FSA individual are updated simultaneously. The main disadvantage of this process is the principle that different food habitats are dependent and a change in one food habitat can have a negative effect on others, preventing them from finding optimal variables in their own habitats.

To further enhance the intensive search of each foraging area, the following steps are performed for each iteration. The newly generated solution is produced:

$$FS_{best,j}^{new} = Cx\left(FS_{best,j}^{old}, En, He \right),\ j = 1,2,\ldots,n. \quad (2.31)$$

Step 14. Training of FSA knowledge bases.

In the mentioned research, the learning method based on artificial neural networks evolving for changes in the nature of movement of each FSA for more accurate analysis results in the future.

Step 15. Determining the amount of necessary computing resources, intelligent decision-making support system.

In order to prevent looping of calculations on Steps 1–14 of the method and to increase the efficiency of calculations, the system load is additionally determined. If the specified threshold of computational complexity is exceeded, the amount of software and hardware resources that must be additionally involved is determined using the method proposed in the work [27].

The end of algorithm.

A method of finding solutions using an improved algorithm of flying squirrels is proposed. In order to evaluate the effectiveness of the developed method, its comparative evaluation was carried out based on the results of research presented in works [3–6, 23, 24].

Simulation of the solution search processing method was carried out in accordance with Steps 1–15. Simulation of the work of the proposed method was carried out in the MathCad 14 software environment (USA). The assessment of the elements of the operational situation of the group of troops (forces) was the task that was solved during the simulation.

The performance of the improved FSA is compared with the following natural optimization algorithms, including the basic FSA, the fruit fly optimization algorithm, the improved fruit fly optimization algorithm and the cloud model-based fruit fly flight optimization algorithm. The comparison was made with unimodal and multimodal functions. Each function is calculated for ten independent runs to better compare the results of different algorithms.

The results of the comparative analysis are shown in **Table 2.5**.

Table 2.5 Results of comparative analysis

Parameter	Advanced FSA	Basic FSA	Particle swarm algorithm	Flight optimization algorithm of the fruit fly based on the cloud model	Improved fruit fly optimization algorithm	Fruit fly optimization algorithm
$Iter_{max}$	10000	10000	10000	10000	10000	10000
NP	50	50	50	50	50	50
G_c	1.9	1.91	–	–	–	–
sf	18	18	–	–	–	–
P_{dpmax}	0.12	–	–	–	–	–
P_{dpmin}	0.001	–	–	–	–	–
P_{dp}	–	0.15	–	–	–	–
C_1 and C_2	–	–	2.1	–	–	–
w	–	–	0.91	–	–	–
En_max	–	–	–	(UB–UL)/4	–	–
λ_{max}	–	–	–	–	(UB–UL)/2	–
λ_{min}	–	–	–	–	0.000014	–
$randValue$	–	–	–	–	–	1

CHAPTER 2

The initial data for the comparative analysis of the effectiveness of the proposed FSA are set the same for all algorithms, for example, the population size $NP=50$; the maximum number of iterations $Iter_{max}=10\,000$. The specific parameters of the algorithm are chosen in the same way. **Table 2.1** summarizes both general and specific parameters for the improved FSA and the other five algorithms. The error value, defined as $(f(x)-F_{min})$, is written for the solution x, where $f(x)$ is the optimal fitness value of the function calculated by the algorithms and F_{min} is the true minimum value of the function. The mean and standard deviation of the error values for all independent runs are calculated.

As a result of the simulation, sets of input parameters were obtained that ensure the optimal operation of the algorithm under the given conditions (**Table 2.6**).

Table 2.6 Results of the AIP algorithm for test functions

Function	The dimensionality of the function	Number of AIP	Number of iterations	Result	Average time, sec.
De Jong (global optimum: 0)	2	4	10000	0	0
	5	14		0	0.1
	10	28		0.001	3.17
	30	30		0.007	30.42
Rastrigin (global optimum: 0)	2	6	10000	0	0
	5	64		0	18.23
	10	50		0.03	62.5
	30	50		0.97	528.4
Hryvnoka (global optimum: 0)	2	6	10000	0	0
	5	16		0.002	0.17
	10	30		0.004	4.77
	30	43		0.028	89.38
Ackley (global optimum: 0)	2	6	10000	0	0
	5	24		0.001	0.15
	10	42		0.013	3.24
	30	50		0.021	66.73
Bukin (global optimum: 0)	2	8	10000	0	0
	5	20		0.002	2.08
	10	40		0.03	4.16
	30	50		0.85	70.4

Analyzing the performance results of the improved algorithm, shown in **Table 2.6**, it can be seen that for functions with a small number of parameters, the algorithm demonstrates its greatest

efficiency, however, when the dimension of multi-extremal functions with a complex landscape (such as the Rastrigin, Hryvnoka, Bukin functions) increases, there is a slight deviation from global optimum, this deviation can be smoothed out by increasing the number of iterations and agents that affect the duration of the method.

The Rosenbrock function should be noted separately: when the number of parameters increases to more than 10, the FSA shows a rather noticeable discrepancy from the optimal solution, so to achieve the required accuracy, a serious increase in time costs is required, which makes the method ineffective in this particular case.

It can be seen that the improved FSA is able to converge to the true value for most unimodal functions with the fastest convergence speed and the highest accuracy, while the convergence results of the particle swarm algorithm and the fruit fly algorithm are far from satisfactory.

Based on the research conducted, it can be said that FSA is more effective for working with functions with a small number of parameters, however, one of the ways to improve the accuracy of the solutions found for multiparameter multimodal functions is to modify or hybridize the method with other algorithms.

The advantages of the proposed method are due to the following:

– at the initial display of FSA, the type of uncertainty is taken into account during their search (Step 2);

– additional determination of the suitability of the FSA search location, which reduces the time to search for a solution (Step 5);

– universality of strategies for finding food areas, which allows to classify the type of data to be processed (Steps 6, 7);

– taking into account the presence of a predator, which allows to avoid local optima (Steps 8, 9);

– universality of solving the task of analyzing the state of FSA objects due to the hierarchical nature of their description (Steps 1–15);

– possibility of quick search for solutions due to the simultaneous search for a solution by several FSA (Steps 1–15, Table 2.5 and 2.6);

– adequacy of the obtained results (Steps 1–15);

– adaptive change of the search area by individual FSA (Step 13);

– ability to avoid the local extremum problem (Steps 1–15);

– possibility of in-depth learning of FSA knowledge bases (Step 15).

The disadvantages of the proposed method include:

– loss of informativeness while assessing the state of the object of analysis due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of the object of analysis;

– loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the state of the heterogeneous object of analysis;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous object of analysis;

– to reduce the use of computing resources of decision-making support systems.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proving information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

2.3 THE DEVELOPMENT OF A METHOD FOR ASSESSING THE STATE OF DYNAMIC OBJECTS USING A POPULATION ALGORITHM

The proposed approach is a population swarm algorithm based on the simulation of mating behavior of snakes, capable of providing solutions to optimization problems in a multiple iterative process based on search capabilities in the problem-solving space.

The method of assessing the state of dynamic objects using the population algorithm consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available on the dynamic object to be evaluated are entered. Snakes enter their mating season when it's cold outside and they can find food.

Step 2. Initial placement of agents on the search plane. The snake optimization algorithm (SOA) is initialized by generating a random population according to the equation (2.32). The population of snake agents (SA) is then divided into two groups, male and female (2.33):

$$x_{i,j} = Lb_j + r \cdot \left(Ub_j - Lb_j\right),\ i = 1,2,\ldots,N,\ j = 1,2,\ldots,m, \tag{2.32}$$

$$N_{female} \cong \frac{N}{2},\ N_{male} = N - N_{female}, \tag{2.33}$$

where x_{ij} is the dimension of the i-th SA; m is the number of measurements; N is the population size of the SA; r is the degree of uncertainty of information about the state of the object; Ub and Lb are the upper and lower limits of the j-th dimension, respectively. N_{female} is the number of SA females in the population, and N_{male} is the number of SA males in the population.

The best value obtained for the objective function corresponds to the best member of the swarm (the best possible solution) and the worst value obtained for the objective function corresponds to the worst member of the population (the worst possible solution). Since at each iteration the position of the population SA in the problem solution space is updated, the best population SA must also be updated based on the comparison of the updated values for the objective function. At the end of the implementation of the algorithm, the position of the best SA of the population, obtained during the iterations of the algorithm, is presented as a solution to the task of finding a solution to the state of the dynamic object.

Step 3. Numbering of SA in the population, i, $i \in [0, S]$. At this stage, each SA of the population is assigned a serial number.

Step 4. Determining the initial velocity of population SA.

Initial speed v_0 each SA of the population is determined by the following expression:

$$v_i = \left(v_1, v_2 \ldots v_S\right),\; v_i = v_0. \tag{2.34}$$

In the planning of the proposed approach, the position of the SA population in the problem-solving space is updated based on the simulation of exploitation, struggle and mating strategies.

Step 5. Preliminary assessment of the SA search area. In this procedure, the search area in natural language is determined precisely by the halo of the existence of SA. Given that the sources of food in SA are diverse, let's sort out the quality of the food.

Step 6. Classification of food sources for SA.

The attractiveness of food significantly depends on the temperature indicators (T) of the surrounding environment and the quality of food is calculated according to equations (2.35) and (2.36):

$$T = \exp\left(\frac{-t}{t_{\max}}\right), \tag{2.35}$$

$$FQ = c_1 \times \exp\left(\frac{t - t_{\max}}{t_{\max}}\right), \tag{2.36}$$

where t is the number of the current iteration; $t_{\max}$ is the total number of iterations; c_1 is the food attractiveness coefficient for each temperature range.

In this algorithm, SA choose a place to search for food, according to c_1, then SA update the position. Research behavior of SA males and females is expressed mathematically in equations (2.37) and (2.38), respectively:

$$x_{i,j}\left(t+1\right) = x_{k,j}\left(t\right) \mp c_2 \times A_{i,male}\left(\left(Ub - Lb\right) \times r_1 + Lb\right),\; A_{i,male} = \exp\left(\frac{-F_{r,male}}{F_{i,male}}\right), \tag{2.37}$$

$$x_{i,j}(t+1) = x_{k,j}(t+1) \mp c_2 \times A_{i,female} A_{i,male} \left((Ub - Lb) \times r_1 + r_2 + Lb \right),$$
$$A_{i,female} = \exp\left(\frac{-F_{r,female}}{F_{i,female}} \right), \tag{2.38}$$

where k is a random integer in the range $(1, N/2)$; $x_{k,j}$ is a random value of the number of males/females in the SA population; and r_1 and r_2 are random numbers in the range (0, 1). $A_{i,female}$ and $A_{i,male}$ are the ability of SA males and females to find food; $F_{r,male}$ is the fitness of a pre-selected random SA male; $F_{r,female}$ is the fitness of a pre-selected random female SA. $F_{i,male}$ and $F_{i,female}$ are the i-th indicator of the male and female SA, respectively. The direction operator (±) scans all possible directions randomly in the given search space.

Step 7. Checking the fulfillment of the condition. If $FQ > T$ is the transition to operation mode. If $FQ < T$ is the return to reconnaissance mode (Step 6).

Step 8. Operation mode.

Step 8.1. Checking the fulfillment of the condition.

If $T > 0.6$ (hot), SA will move towards food according to equation (2.38):

$$x_{i,j}(t+1) = x_f \mp c_3 \times T \times r_3 \times \left(x_f - x_{i,j}(t) \right), \tag{2.38}$$

where $x_{i,j}$ are the locations of male and female SA; x_f are the best SA, c_3 is a constant equal to 2; and r_3 is a random number in the range (0,1).

Step 8.2. Checking the fulfillment of the condition.

If $FQ < T$ ($T < 0.6$) (cold), SA fight or mate.

Step 8.3. SA combat.

The fighting abilities of male SA F_{male} and female SA F_{female} can be written in equations (2.39) and (2.40):

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,male} \times r_4 \times \left(x_{best,female} - x_{i,male}(t) \right), \; F_{i,male} = \exp\left(\frac{-F_{best,f}}{F_i} \right), \tag{2.39}$$

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,female} \times r_5 \times \left(x_{best,male} - x_{i,female}(t+1) \right),$$
$$F_{i,female} = \exp\left(\frac{-F_{best,male}}{F_i} \right), \tag{2.40}$$

where $x_{i,j}$ is the location of male and female SA; $x_{best,female}$ and $x_{best,male}$ are the positions of the best SA in female and male groups, respectively; $F_{i,male}$ is the fight of SA in males and $F_{i,female}$ is the fight of SA in females. In addition, c_4 is the food saturation coefficient of SA; and r_4 and r_4 are random numbers in the range (0, 1).

Step 8.4. Checking the fulfillment of the condition. If $c_4 \leq c_{4nonor}$ – go to Step 8.3, if $c_4 \geq c_{4nonor}$ – go to Step 8.5 – pairing.

Step 8.5. SA pairing. During mating, male and female SA update their positions, according to equations (2.41) and (2.42):

$$x_{i,male}(t+1) = x_{i,m}(t) \pm c_5 \times M_{i,male} \times r_6 \times \left(FQ \times x_{i,female} - x_{i,male}(t)\right),$$

$$M_{i,male} = \exp\left(\frac{-f_{i,female}}{f_{i,male}}\right); \tag{2.41}$$

$$x_{i,female}(t+1) = x_{i,f}(t) \pm c_5 \times M_{i,female} \times r_7 \times \left(FQ \times x_{i,male} - x_{i,female}(t+1)\right),$$

$$M_{i,female} = \exp\left(\frac{-f_{i,male}}{f_{i,female}}\right), \tag{2.42}$$

where $x_{i,m}$ and $x_{i,f}$ are the positions of the i-th male and female SA; $M_{i,male}$ and $M_{i,female}$ are the mating abilities of the male and female SA; c_5 is the fertility rate of the new SA; r_6 and r_7 are random numbers in the range (0, 1).

Step 8.6. Checking the fulfillment of the condition. If $c_5 \leq c_{5nonor}$ – go to Step 8.5, if $c_5 \geq c_{5nonor}$ – go to Step 8.7 – replacement of SA in the population.

Step 8.7. Replacement of SA in the population. In this step, the replacement of SA that do not meet the requirements for the fertility of SA takes place, according to expressions (2.43), (2.44):

$$x_{w,male} = Lb + r_8 \times (Ub - Lb), \tag{2.43}$$

$$x_{w,female} = Lb + r_8 \times (Ub - Lb), \tag{2.44}$$

where $x_{w,male}$ is the worst male SA, while $x_{w,female}$ is the worst female SA; r_8 is a random number in the range (0, 1).

Step 9. Checking the stop criterion. The algorithm terminates when the maximum number of iterations is completed. Otherwise, the behavior of generating new places and checking conditions is repeated.

Step 10. Learning SA knowledge bases.

In the research, the learning method based on evolving artificial neural networks developed in the research [2] is used to learn the knowledge bases of each SA. The method is used to change the nature of movement of each SA, for more accurate analysis results in the future.

Step 11. Determining the amount of necessary computing resources, intelligent decision-making support system.

In order to prevent looping of calculations on Steps 1–10 of this method and to increase the efficiency of calculations, the system load is additionally determined. When the specified threshold of computational complexity is exceeded, the amount of software and hardware resources that must be additionally involved is determined using the method proposed in the work [31].

The end of algorithm.

The effectiveness of the method of assessing the state of dynamic objects using the population algorithm is compared with the help of functions – the form of which is given in **Tables 2.7**, **2.8**.

Table 2.7 Unimodal functions and their parameters, according to which the simulation was carried out

Function name	Range	f_{min}	Dim
$f1(x)=\sum_{i=2}^{n} x_i^2$	[–300, 300]	0	26
$f2(x)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$		0	26
$f3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$		0	26
$f4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le n\}$		0	26
$f5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1^2\right)\right]$		0	26
$f6(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$		0	26
$f7(x)=\sum_{i=1}^{n} ix_i^4+\text{rand}\left[0,1\right)$		0	26
$f8=\sum_{i=1}^{d} -x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$		–306.194 ×dim	26

Table 2.8 Multimodal functions and their parameters according to which modeling was carried out

Function name	Range	f_{min}	Dim
$f9=\sum_{i=1}^{d}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	[–300, 300]	0	26
$f10(x)=\sum_{i=1}^{n} -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$		0	26
$f11(x)=\frac{1}{4}\times10^{-3}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}+1\right)$		0	26
$f12(x)=\frac{\pi}{n}\left\{\begin{matrix}10\sin\left(\pi y_1\right)+\\+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]+\left(y_n-1\right)^2\end{matrix}\right\}+\sum_{i=1}^{n} u\left(x_i,10,100,4\right);$ $y_i=\frac{x_i+5}{4}u\left(x_i,a,k,m\right)=\left\{\begin{matrix}k\left(x_i-a\right)^m & Egerx_i>a\\ k\left(-x_i-a\right)^m & Egerx_i<a\\ 0 & diger\end{matrix}\right\}$		0	26

From the analysis of **Tables 2.7**, **2.8**, it can be concluded that the proposed method ensures stable operation of the algorithm for the main test functions of unimodal and multimodal type.

CHAPTER 2

In order to obtain a comparative assessment based on the criterion of information processing efficiency at a given assessment reliability, a comparative assessment of the proposed method with swarm algorithms was carried out (**Tables 2.9**, **2.10**).

Table 2.9 Comparative evaluation of the proposed method with swarm algorithms according to the criterion of efficiency of information processing

Test function number	Algorithm of a flock of gray wolves	Algorithm of a flock of walruses	Particle swarm algorithm	The proposed method
F1	3.56E-195	2.40E-195	1.10E-195	2.24E-196
F2	2.57E-97	2.45E-98	2.07E-98	2.01E-98
F3	2.54E-128	1.14E-127	1.63E-123	1.51E-127
F4	7.00E-87	7.01E-87	4.24E-87	5.38E-87
F5	1.57E+01	1.52E+01	1.50E+01	1.57E+01
F6	2.61E-02	1.31E-03	1.45E-03	2.25E-03
F7	1.03E-04	9.17E-05	1.09E-04	1.04E-04
F8	–1,25E+04	**–1,26E+04**	**–1,26E+04**	–1,25E+04
F9	4.73E-01	1.28E-01	**9.09E-02**	5.62E-01
F10	4.09E-15	3.97E-15	3.97E-15	3.97E-15
F11	8.10E-03	3.42E-03	**1.21E-03**	2.73E-03
F12	2.09E-02	3.56E-02	2.15E-02	2.1E-02

Table 2.10 Comparative evaluation of the proposed method with swarm algorithms according to the criterion of efficiency of information processing

Test function number	Algorithm of a flock of monkeys	Algorithm of flock of hawks	The bat swarm algorithm	Algorithm of a herd of coots	The proposed method
F1	5.78E-194	**2.16E-196**	1.40E-195	3.67E-196	2.12E-196
F2	1.26E-98	1.37E-98	3.45E-98	2.28E-98	**1.01E-98**
F3	**2.22E-128**	1.05E-124	5.98E-125	1.06E-125	1.00E-127
F4	1.23E-86	5.58E-87	**2.23E-87**	6.91E-87	2.31E-87
F5	1.52E+01	1.43E+01	1.69E+01	1.46E+01	**1.30E+01**
F6	2.00E-02	1.91E-03	5.65E-03	2.22E-03	**1.10E-03**
F7	9.53E-05	1.21E-04	1.14E-04	1.04E-04	**2.15E-05**
F8	**–1,26E+04**	**–1,26E+04**	–1,25E+04	–1,25E+04	**–1,26E+04**
F9	9.05E-01	5.98E-01	1.15E+00	4.62E-01	2.70E-01
F10	4.09E-15	3.85E-15	3.85E-15	3.97E-15	**3.61E-15**
F11	1.13E-02	1.34E-03	4.22E-03	3.71E-03	1.50E-03
F12	**3.67E-04**	9.56E-03	4.30E-03	9.35E-03	3.22E-02

As can be seen from the **Tables 2.7–2.10** increasing the efficiency of decision making is achieved at the level of 13–19 % due to the use of additional procedures.

It can be seen that the state estimation method of dynamic objects using the population algorithm is able to converge to the true value for most unimodal and multimodal functions with the fastest convergence speed and the highest accuracy, while the convergence results of the monkey flock algorithm and the gray wolf flock algorithm are not satisfactory for multimodal functions.

The advantages of the proposed method are due to the following:

– initial position of the SA is carried out taking into account the type of uncertainty, in comparison with works [9, 14, 21];

– initial speed of each SA is taken into account, in comparison with works [9–15];

– suitability of the place of search for SA is determined, which reduces the time of searching for a solution (Step 5), in comparison with works [14, 16, 17];

– universality of strategies for finding places of food in SA, which allows to classify the type of data to be processed (Step 6), in comparison with works [14, 16, 17];

– the possibility of adjusting the speed of the SA movement by adjusting the temperature of the surrounding environment, which achieves the determination of the priority of finding a solution in a certain plane, in comparison with works [11, 13, 17–19];

– possibility to explore the solution spaces of functions described by atypical functions, due to the use of operation mode procedures (Step 8), in comparison with works [9, 12, 13–18];

– possibility of flexible regulation of the transition from the fighting mode to the mating mode of SA due to the use of the food saturation coefficient, in comparison with works [9–23];

– to replace persons, unfit for search by using the fertility rate of SA (Step 8.7), in comparison with works [9, 12, 13–18];

– possibility of simultaneously searching for a solution in different directions (Steps 1–11, **Tables 2.7–2.10**);

– possibility of in-depth learning of SA knowledge bases (Step 10), in comparison with works [9–23];

– possibility of calculating the necessary amount of computing resources, which must be involved in case of impossibility of carrying out calculations with available computing resources (Step 11), in comparison with works [9–23].

The disadvantages of the proposed method include:

– loss of informativeness while assessing the state of complex dynamic objects due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of complex dynamic objects;

– loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the condition of complex dynamic objects;

– to determine effective measures to increase the efficiency of management of complex dynamic objects;

– to increase the speed of assessment of the state of complex dynamic objects;

– to reduce the use of computing resources of decision-making support systems.

A limitation of the research is the need to have an initial condition database complex dynamic object, the need to take into account the time delay for collecting and proving information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

The research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 21–23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

CONCLUSIONS

1. The method implementation algorithm is determined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noise;

– to implement adaptive strategies for finding food sources for AG;

– to take into account the presence of a predator while choosing food sources by swarm agents of the combined swarm algorithm;

– to take into account the available computing resources of the state analysis system of complex dynamic objects of AG;

– to change the search area by individual AG;

– to change the speed of AG movement;

– to take into account the priority of search by swarm agents of the combined swarm algorithm;

– to carry out the initial exposure of the AG taking into account the type of uncertainty;

– to determine the best AG using an improved genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of data on the state of complex dynamic objects;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of complex dynamic objects due to the hierarchical nature of the description;

– to check the adequacy of the obtained results;

– to combine various swarm algorithms for mutual verification of the adequacy and reliability of the obtained results;

– to avoid the problem of local extremum.

2. An example of the use of the proposed method has been carried out on the example of solving the task of determining the composition of an operational group of troops (forces) and elements of its operational construction. The specified example showed a 10–12 % increase in the efficiency of data processing due to the use of additional improved procedures.

3. The method implementation algorithm has been defined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noisy data;

– to implement adaptive strategies for finding food sources;

– to take into account the presence of a predator while choosing food sources;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to change the search area by individual FSA;

– to take into account the priority of FSA search;

– to carry out the initial exposure of FSA individuals, taking into account the type of uncertainty;

– to carry out accurate training of FSA individuals;

– to determine the best FSA individuals using a genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

4. Conducted example of using the proposed method on the example of assessing and forecasting the state of the operational situation of a group of forces. The specified example showed an increase in the efficiency of data processing at the level of 21–25 % due to the use of additional improved procedures of adding correction coefficients for uncertainty and noisy data, selection and training of FSA.

5. The method implementation algorithm is defined, thanks to additional and improved procedures, which allows:

– to determine the initial position of the SA, taking into account the type of uncertainty due to the use of a correction factor for the degree of awareness of the state of the initial situation in relation to the analysis object;

– to take into account the initial speed of each SA, which allows to investigate functions that are complex in terms of volume;

– to ensure the universality of strategies for finding places to eat in SA, which allows to classify the type of data to be processed;

– to regulate the speed of movement of the SA by adjusting the temperature of the surrounding environment, which is achieved by determining the priority of finding a solution in a certain plane;

– to explore the solution spaces of functions described by non-typical functions, due to the use of operation mode procedures;

– flexibly to regulate the transition from the fighting mode to the SA mating mode due to the use of the food saturation coefficient;

– to replace persons unsuitable for search due to the use of the SA fertility rate;

– to carry out a simultaneous search for a solution in different directions, due to changing the temperature of the surrounding environment and adjusting the food saturation coefficient;

– to conduct in-depth training of SA knowledge bases, due to the use of the proposed training method.

6. An example of the use of the proposed method has been carried out on the example of solving the task of determining the composition of an operational group of troops (forces) and elements of its operational construction. The specified example showed a 13–19 % increase in the efficiency of data processing due to the use of additional improved procedures.

REFERENCES

1. Shyshatskyi, A. V., Bashkyrov, O. M., Kostyna, O. M. (2015). Rozvytok intehrovanykh system zviazku ta peredachi danykh dlia potreb Zbroinykh Syl. Ozbroiennia ta viiskova tekhnika, 1 (5), 35–40.
2. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskyi, R., Repilo, I. et al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 37–47. https://doi.org/10.15587/1729-4061.2020.203301
3. Sova, O., Shyshatskyi, A., Salnikova, O., Zhuk, O., Trotsko, O., Hrokholskyi, Y. (2021). Development of a method for assessment and forecasting of the radio electronic environment. EUREKA: Physics and Engineering, 4, 30–40. https://doi.org/10.21303/2461-4262.2021.001940
4. Pievtsov, H., Turinskyi, O., Zhyvotovskyi, R., Sova, O., Zvieriev, O., Lanetskii, B., Shyshatskyi, A. (2020). Development of an advanced method of finding solutions for neuro-fuzzy expert systems of analysis of the radioelectronic situation. EUREKA: Physics and Engineering, 4, 78-89. https://doi.org/10.21303/2461-4262.2020.001353
5. Zuiev, P., Zhyvotovskyi, R., Zvieriev, O., Hatsenko, S., Kuprii, V., Nakonechnyi, O. et al. (2020). Development of complex methodology of processing heterogeneous data in intelligent

decision support systems. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 14–23. https://doi.org/10.15587/1729-4061.2020.208554

6. Shyshatskyi, A., Zvieriev, O., Salnikova, O., Demchenko, Ye., Trotsko, O., Neroznak, Ye. (2020). Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering, 9 (4), 5583–5590. https://doi.org/10.30534/ijatcse/2020/206942020
7. Yeromina, N., Kurban, V., Mykus, S., Peredrii, O., Voloshchenko, O., Kosenko, V. et al. (2021). The Creation of the Database for Mobile Robots Navigation under the Conditions of Flexible Change of Flight Assignment. International Journal of Emerging Technology and Advanced Engineering, 11 (5), 37–44. https://doi.org/10.46338/ijetae0521_05
8. Shyshatskyi, A., Stasiuk, T., Odarushchenko, E., Berezanska, K., Demianenko, H. (2023). Method of assessing the state of hierarchical objects based on bio-inspired algorithms. Advanced Information Systems, 7 (3), 44–48. https://doi.org/10.20998/2522-9052.2023.3.06
9. Ko, Y.-C., Fujita, H. (2019). An evidential analytics for buried information in big data samples: Case study of semiconductor manufacturing. Information Sciences, 486, 190–203. https://doi.org/10.1016/j.ins.2019.01.079
10. Ramaji, I. J., Memari, A. M. (2018). Interpretation of structural analytical models from the coordination view in building information models. Automation in Construction, 90, 117–133. https://doi.org/10.1016/j.autcon.2018.02.025
11. Pérez-González, C. J., Colebrook, M., Roda-García, J. L., Rosa-Remedios, C. B. (2019). Developing a data analytics platform to support decision making in emergency and security management. Expert Systems with Applications, 120, 167–184. https://doi.org/10.1016/j.eswa.2018.11.023
12. Chen, H. (2018). Evaluation of Personalized Service Level for Library Information Management Based on Fuzzy Analytic Hierarchy Process. Procedia Computer Science, 131, 952–958. https://doi.org/10.1016/j.procs.2018.04.233
13. Chan, H. K., Sun, X., Chung, S.-H. (2019). When should fuzzy analytic hierarchy process be used instead of analytic hierarchy process? Decision Support Systems, 125, 113114. https://doi.org/10.1016/j.dss.2019.113114
14. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. Future Generation Computer Systems, 91, 620–633. https://doi.org/10.1016/j.future.2018.06.046
15. Gödri, I., Kardos, C., Pfeiffer, A., Váncza, J. (2019). Data analytics-based decision support workflow for high-mix low-volume production systems. CIRP Annals, 68 (1), 471–474. https://doi.org/10.1016/j.cirp.2019.04.001
16. Harding, J. L. (2013). Data quality in the integration and analysis of data from multiple sources: some research challenges. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-2/W1, 59–63. https://doi.org/10.5194/isprsarchives-xl-2-w1-59-2013

17. Kosko, B. (1986). Fuzzy cognitive maps. International Journal of Man-Machine Studies, 24 (1), 65–75. https://doi.org/10.1016/s0020-7373(86)80040-2
18. Koval, M., Sova, O., Shyshatskyi, A., Artabaiev, Y., Garashchuk, N., Yivzhenko, Y. et al. (2022). Improving the method for increasing the efficiency of decision-making based on bio-inspired algorithms. Eastern-European Journal of Enterprise Technologies, 6 (4 (120)), 6–13. https://doi.org/10.15587/1729-4061.2022.268621
19. Maccarone, A. D., Brzorad, J. N., Stone, H. M. (2008). Characteristics And Energetics of Great Egret And Snowy Egret Foraging Flights. Waterbirds, 4, 541–549. https://doi.org/10.1675/1524-4695-31.4.541
20. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. https://doi.org/10.15587/1729-4061.2019.180197
21. Mahdi, Q. A., Shyshatskyi, A., Prokopenko, Y., Ivakhnenko, T., Kupriyenko, D., Golian, V. et al. (2021). Development of estimation and forecasting method in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 51–62. https://doi.org/10.15587/1729-4061.2021.232718
22. Petrovska, I., Kuchuk, H. (2023). Adaptive resource allocation method for data processing and security in cloud environment. Advanced Information Systems, 7 (3), 67–73. https://doi.org/10.20998/2522-9052.2023.3.10
23. Braik, M., Ryalat, M. H., Al-Zoubi, H. (2021). A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves. Neural Computing and Applications, 34 (1), 409–455. https://doi.org/10.1007/s00521-021-06392-x
24. Khudov, H., Khizhnyak, I., Glukhov, S., Shamrai, N., Pavlii, V. (2024). The method for objects detection on satellite imagery based on the firefly algorithm. Advanced Information Systems, 8 (1), 5–11. https://doi.org/10.20998/2522-9052.2024.1.01
25. Poliarush, O., Krepych, S., Spivak, I. (2023). Hybrid approach for data filtering and machine learning inside content management system. Advanced Information Systems, 7 (4), 70–74. https://doi.org/10.20998/2522-9052.2023.4.09
26. Chalyi, S., Leshchynskyi, V. (2023). Possible evaluation of the correctness of explanations to the end user in an artificial intelligence system. Advanced Information Systems, 7 (4), 75–79. https://doi.org/10.20998/2522-9052.2023.4.10
27. Raskin, L., Sira, O. (2016). Method of solving fuzzy problems of mathematical programming. Eastern-European Journal of Enterprise Technologies, 5 (4 (83)), 23–28. https://doi.org/10.15587/1729-4061.2016.81292
28. Lytvyn, V., Vysotska, V., Pukach, P., Brodyak, O., Ugryn, D. (2017). Development of a method for determining the keywords in the slavic language texts based on the technology of web mining. Eastern-European Journal of Enterprise Technologies, 2 (2 (86)), 14–23. https://doi.org/10.15587/1729-4061.2017.98750

29. Stepanenko, A., Oliinyk, A., Deineha, L., Zaiko, T. (2018). Development of the method for decomposition of superpositions of unknown pulsed signals using the secondorder adaptive spectral analysis. Eastern-European Journal of Enterprise Technologies, 2 (9 (92)), 48–54. https://doi.org/10.15587/1729-4061.2018.126578
30. Gorbenko, I., Ponomar, V. (2017). Examining a possibility to use and the benefits of post-quantum algorithms dependent on the conditions of their application. Eastern-European Journal of Enterprise Technologies, 2 (9 (86)), 21–32. https://doi.org/10.15587/1729-4061.2017.96321
31. Koval, M., Sova, O., Orlov, O., Shyshatskyi, A., Artabaiev, Y., Shknai, O. et al. (2022). Improvement of complex resource management of special-purpose communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 34–44. https://doi.org/10.15587/1729-4061.2022.266009
32. Mahdi, Q. A., Zhyvotovskyi, R., Kravchenko, S., Borysov, I., Orlov, O., Panchenko, I. et al. (2021). Development of a method of structural-parametric assessment of the object state. Eastern-European Journal of Enterprise Technologies, 5 (4 (113)), 34–44. https://doi.org/10.15587/1729-4061.2021.240178
33. Koval, V., Nechyporuk, O., Shyshatskyi, A., Nalapko, O., Shknai, O., Zhyvylo, Y. et al. (2023). Improvement of the optimization method based on the cat pack algorithm. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 41–48. https://doi.org/10.15587/1729-4061.2023.273786

DOI: 10.15587/978-617-8360-04-7.CH3

Svitlana Kashkevich, Nina Kuchuk, Oleksii Kuvshynov,
Ganna Plekhova, Oleksandr Yefymenko, Andrii Veretnov

CHAPTER 3

THE DEVELOPMENT OF METHODS FOR EVALUATING THE STATE OF COMPLEX TECHNICAL SYSTEMS USING ARTIFICIAL INTELLIGENCE THEORY

ABSTRACT

In this chapter of the research, the methods for assessing the state of complex technical systems using the theory of artificial intelligence are proposed. The basis of this research is the theory of artificial intelligence. The methods aimed at solving optimization tasks, variable solutions are defined in such a way that complex technical systems work at their best point (mode) based on the optimization criteria determined.

In the research, the authors proposed:

– the method of assessing the state of complex technical systems using bio-inspired algorithms;

– the method of finding solutions using the population algorithm of global search optimization;

– the method of finding solutions using the improved algorithm of shoals of fish;

– the method of finding solutions using an improved algorithm of jumping frogs.

Each of the methods was based on canonical optimization algorithms, but they were improved by the authors of this research.

The essence of the improvement of these methods, which is the scientific novelty of each of them:

– taking into account a priori known coefficient regarding the degree of uncertainty of data about a complex technical system and the coefficient determined during the work of algorithms regarding the noise of the data;

– the procedure of deep training of agents of the flock allows, in the presence of reliable data, to significantly reduce the time for decision making;

– the reliability of decisions is improved due to the selection of swarm agents. Selection in each of the algorithms is carried out using an improved genetic algorithm.

A limitation of the research is the need to have an initial condition database complex technical system, the need to take into account the time delay for collecting and proving information from sources of information extraction.

It is advisable to use the proposed approach to solve the tasks of evaluating complex and dynamic processes characterized by a high degree of complexity.

KEYWORDS

Optimization problems, complex technical systems, multi-agent systems, artificial intelligence, reliability and adequacy.

Metaheuristic algorithms are widely used for optimization in various tasks, in particular for the purpose of selecting informative subsets of features, while building a machine learning model [1–10].

Removing redundant features helps to avoid overtraining the model and reduce its complexity. While using metaheuristics as a feature selection tool, a necessary condition is the ability of the algorithm to search the binary space [11–27]. However, while some metaheuristics, such as the genetic algorithm, were originally designed to work with binary input vectors, others were designed to work in a continuous search domain. It is obvious that the genetic algorithm alone is not capable of being a universal tool for any data, as this contradicts the free breakfast theorem [28–30].

The most famous representative of heuristic methods is swarm intelligence, which describes the collective behavior of a decentralized, self-organizing system [31–36].

There are a large number of swarm algorithms, for example: particle swarm method, ant algorithm, cuckoo, bat, fish, bee algorithm, colonization algorithms, etc. [37–40].

The use of swarm algorithms to find solutions regarding the state of the objects of analysis allows:

– to analyze the stability of the state of heterogeneous objects in the process of combat use (exploitation);

– to analyze the direct, aggregated and mediated interaction of systemic and external factors;

– to assess the reach of target situations of facility management;

– to make a scenario analysis for various destructive effects;

– to forecast the changes in the state of heterogeneous objects under the influence of destabilizing factors during combat use (exploitation);

– to make modeling and analysis of the dynamics of changes in the state of interdependent parameters of heterogeneous objects.

At the same time, the use of metaheuristic algorithms in the canonical form does not allow to obtain an operational assessment of the state of the object with a given reliability. The above mentioned determines the search for new (improvement of existing) approaches to the assessment and forecasting of the state of objects by combining already known swarm algorithms with their further improvement.

3.1 A METHOD OF ASSESSING THE STATE OF COMPLEX TECHNICAL SYSTEMS USING BIO-INSPIRED ALGORITHMS

The method of assessing the condition of complex technical systems using bio-inspired algorithms consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available on the object to be analyzed are entered. The existing model of the analysis object is also initialized. A set of analyzed functions of the state of the object is specified with the implementation of the corresponding procedures, in this case, two functions are implemented:

$$F(x)=\sin(x), \text{ and } F(x)=-\left(x^2+12x-21\right).$$

Step 2. Processing of raw data taking into account the degree of uncertainty.

At this stage, the type of uncertainty is taken into account about the object to be analyzed and initialization of the basic state model of the object to be analyzed [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. This is done with the help of correction coefficients.

The list of variables used, with their designations inside the algorithm and a description of their purpose, is presented in the **Table 3.1**.

Step 3. Numbering of bat agents, i, $i \in [0, S]$.

Step 4. Placement of bat agents (BA) taking into account the degree of uncertainty about the state of the analysis object in the search space:

$$x \in \left(x_{\min}, x_{\max}\right),\ x_i=\left(x_1, x_2 \ldots x_S\right), \tag{3.1}$$

$$x=x_{\min}+\left(x_{\min}-x_{\max}\right)\cdot \iota(\ ), \tag{3.2}$$

where $x_{\min}$, $x_{\max}$ are the minimum and maximum value of the search area; ι is the degree of uncertainty about the state of the analysis object. At the same time, the function $\iota(\ )$ returns values in the interval [0; 1].

Step 5. Setting the initial speed of the BA and the echolocation frequency of each BA.

Initial speed v_0 of each BA is defined by the following expression:

$$v_i=\left(v_1, v_2 \ldots v_S\right),\ v_i=v_0. \tag{3.3}$$

The initial frequency of BA is determined by the expression:

$$w_i=\left(w_1, w_2 \ldots w_S\right),\ w_i=w_{\min}, \tag{3.4}$$

where w_i is the value of the BA frequency with number i; $w_{\min}$ is the minimum BA frequency.

If $i<S$, then return to Step 4.

Step 6. Finding the best BA.

The substitution of the best value of BA x_i in the expression of the analyzed function $F(X)$. The BA value closest to the extremum (in this case, to the maximum) is considered the best,

$x_k^* = \max(F(x))$, where k is the number of the best BA. To find the best BA, the improved genetic algorithm developed in the work [22] is used.

Step 7. BA migration. Migration of agents is performed: agents are moved one step according to the migration procedure and iterations are performed with n, where $n \in [0, N]$.

Step 8. Repeated numbering of BA i, $i \in [0, S]$.

Step 9. Changing the search parameters.

New values of position, speed and frequency for BA are calculated – x_i^l, v_i^l and w_i^l. The frequency is modified taking into account the degree of data noise according to the following expression:

$$w_i^l = w_{\min} + (w_{\max} - w_{\min}) \cdot \eta(\ ). \tag{3.5}$$

Step 10. Changing the speed of BA movement.

Speed modification is performed:

$$v_i^l = v_i + w_i (x^* - x_i), \tag{3.6}$$

where $(x^* - x_i)$ is the approximation of all BA by $\eta \to \max$.

Step 11. BA movement.

BA is moved in accordance with the formula: $x_i^l = x_i + v_i$. Checking the condition $i<S$, if $i<S$, go to Step 10.

Step 12. Checking the condition for starting the local search procedure.

If $E_r > \eta(\ )$, then the local search procedure is started (go to Step 16), otherwise, the transition to Step 19 is performed. The search procedure around the best solution is carried out with the probability E_r. The resulting decision is applied as a new current provision of BA x_i.

Step 13. Checking the number of iterations of the solution search n. If the search is not performed for the first time, then go to Step 19. Otherwise: the BA volume is calculated according to the formula:

$$a_i = (a_1, a_2 \ldots a_S);\ a_i^l = a_{\min} + (a_{\max} - a_{\min}) \cdot \eta(\ ), \tag{3.7}$$

where a_i^l is the new volume value and the transition to Step 18 is performed.

Step 14. The average volume of BA is calculated using the following expression:

$$a_{sr} = \left(\sum_{i=0}^{S} a \right) \Big/ S. \tag{3.8}$$

Step 15. Change in the current position of BA:

$$xx_i^l = x_i + a_{sr} \cdot U[-1,1]. \tag{3.9}$$

The function $U[-1, 1]$ returns a random value in the interval $[-1; 1]$.

Step 16. The local search procedure is performed until the BA is closer to the search goal: $F(xx_i) > F(x_i)$.

Step 17. Limitation of the search area:

$$xx_i = \max\{x_{\min}, x_i\},\ xx_i = \min\{x_{\max}, x_i\}. \tag{3.10}$$

Step 18. Global search for a solution.

With the probability E_a, a global search procedure is performed in the neighborhood of the current solution for all i-th BA, $i,\ i \in [0, S]$.

Step 19. If $F(xx_i) < F(x_i)$ and $E_a > \eta()$, then a new decision is made: $x_i = xx_i$.

If $i < S$, then the transition to Step 20 is performed. The search area is limited for all i-th BA, $i,\ i \in [0, S]$.

Step 20. Checking the fulfillment of a set of conditions:

1) if $x_i < x_{\min}$, then: $x_i = x_{\min}$, $V_i = V_0$;

2) if $x_i > x_{\max}$, then: $x_i = x_{\max}$, $V_i = V_0$;

3) if $i < S$, the transition to Step 10 is made;

4) if $n < N$, then Step 7 is performed.

After performing all iterations, the value of x^* is taken in relation to the state of the analysis object.

Step 21. Learning knowledge bases.

In the research, the learning method based on evolving artificial neural networks, developed in the research [2], is used for training knowledge bases.

The end of algorithm.

The proposed method of assessing the state of complex technical systems using bio-inspired algorithms. To assess the effectiveness of the developed method, it was compared with the results of research presented in works [3–6, 23, 24, 37].

The simulation of the solution search processing method was carried out in accordance with expressions (3.1)–(3.10). The proposed method was simulated in the MathCad 14 software environment (USA). The assessment of elements of the operational situation of the group of troops (forces) was the task to be solved during the simulation.

Initial data for assessing the state of the operational situation using the improved method:

– the number of sources of information about the state of the monitoring object is 3 (radio monitoring tools, remote sensing of the earth and unmanned aerial vehicles) To simplify the modeling, the same number of each tool was taken – 4 tools each;

– the number of informational signs by which the state of the monitoring object is determined – 12. Such parameters include: ownership, type of organizational and staff formation, priority, minimum width along the front, maximum width along the front. The number of personnel, minimum depth along the flank, maximum depth along the flank, the number of samples of weapons

and military equipment (WME), the number of types of WME samples and the number of communication devices), the type of operational construction are also taken into account;

– the variants of organizational and personnel formations are company, battalion, brigade.

Tables 3.1, **3.2** show the dependence of the time of the algorithm on the number of iterations and the size of the population, and the dependence of the absolute error of the algorithm on the number of iterations and the size of the population.

Table 3.1 Dependence of algorithm running time on the number of iterations and population size

No.	The number of iterations	Function	Population size					
			5	10	15	20	50	100
1	10	$f(x)=\sin(x)$	2.88	3.23	3.82	3.71	2.84	3.03
		$f(x)=-x^2+12x-21$	2.85	3.12	2.98	2.99	3.01	3.00
2	20	$f(x)=\sin(x)$	5.15	5.70	6.32	6.34	5.36	7.43
		$f(x)=-x^2+12x-21$	5.26	5.22	5.12	5.19	5.37	5.09
3	50	$f(x)=\sin(x)$	11.97	12.10	12.38	12.01	12.62	13.58
		$f(x)=-x^2+12x-21$	12.27	12.08	12.22	12.30	12.36	13.50
4	100	$f(x)=\sin(x)$	24.39	42.08	31.72	37.52	118.08	214.38
		$f(x)=-x^2+12x-21$	24.37	28.25	39.53	53.35	77.33	151.18

Table 3.2 Dependence of the absolute error of the algorithm on the number of iterations and the size of the population

No.	The number of iterations	Function	Population size					
			5	10	15	20	50	100
1	10	$f(x)=\sin(x)$	0.0239	0.0507	0.0017	0.0012	0.00026	0.00018
		$f(x)=-x^2+12x-21$	0.0006	0.1112	0.01111	0.1114	0.00005	0.0008
2	20	$f(x)=\sin(x)$	0.0079	0.0017	0.00063	0.00083	0.00007	0.00019
		$f(x)=-x^2+12x-21$	0.0085	0.01131	0.0011	0.11131	0.00004	0.00001
3	50	$f(x)=\sin(x)$	0.0004	0.00011	0.00058	0.00042	0.00026	0.00002
		$f(x)=-x^2+12x-21$	0.1115	0.01113	0.10111	0.00000	0	0.11111
4	100	$f(x)=\sin(x)$	0.0008	0.00006	0.000041	0.00001	0	0
		$f(x)=-x^2+12x-21$	0.0111	0	0	0.01111	0.00011	0

Finally, it should be noted that the presented implementation of the bat algorithm showed good performance and the possibility of adjusting the algorithm parameters to change the quality of the obtained results in solving the search optimization problems of object functions.

Table 3.3 presents the comparative results of evaluating the efficiency of learning evolving artificial neural networks.

Table 3.3 Comparative results of the evaluation of the efficiency of learning of evolving artificial neural networks

The system	Algorithm parameters	XB (Xi-Beni Index)	Time, sec
FCM (Fuzzy C-Means)	–	0.2004	2.15
EFCM	Dthr=0.30	0.1018	0.155
EFCM	Dthr=0.23	0.1062	0.2
The proposed system (batch mode)	delta=0.1	0.08	0.2
The proposed system (online mode)	delta=0.1	0.078	0.19

Before training, the features of the observations were normalized to the interval [0, 1].

The research showed that the specified training procedure provides an average of 10–18 % higher training efficiency of artificial neural networks and does not accumulate errors during training (**Table 3.3**).

The indicated results can be seen from the results in the last lines of the **Table 3.3**, as the difference of the Xi-Beni index. At the same time, as already mentioned, in the course of work, known methods accumulate errors, which is why the proposed method proposes the use of evolving artificial neural networks.

The results of the comparative evaluation according to the criterion of efficiency of evaluation are shown in the **Table 3.4**.

Table 3.4 Results of solving the problem

No. of iterations	Method of branches and boundaries [17]	Genetic algorithm [12]	Canonical algorithm of bats [23]	Improved bat algorithm
N	***T*, s**	***T*, s**	***T*, s**	***T*, s**
5	1.125	1.125	1.125	1.114
10	0.625	0.625	0.625	0.600
15	48.97	58.20	58.28	57.71
20	106.72	44.29	43.75	46.95
30	–0.1790	–0.0018	–0.0002	–0.0001
40	–0.158	–0.070	–0.069	–0.049
50	97.76	–974.30	–3.72	–334.11
100	–133.28	–195.71	–196.24	–193.04
200	**7980.89**	**7207.49**	**7198.43**	**7036.48**

As can be seen from the **Table 3.4**, the gain of the specified method of finding solutions is from 11 to 15 % according to the criterion of speed of data processing.

The main advantages of the proposed method are:

– it has a flexible hierarchical structure of indicators, which allows to reduce the task of multi-criteria evaluation of alternatives to one criterion or using a vector of indicators for selection;

– the unambiguousness of the obtained assessment of the state of the analysis object;

the universality of application due to adaptation of the system of indicators during work;

– it does not accumulate learning error due to the use of the learning procedure;

– the possibility of comprehensive learning of the architecture and parameters of artificial neural networks;

– taking into account the type of uncertainty of the initial data while building models of a heterogeneous analysis object;

– the possibility of finding a solution in several directions;

– high reliability of the obtained solutions while searching for a solution in several directions;

– the absence of falling into the local optimum trap.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proving information from intelligence sources.

The advantages of the proposed method are due to the following:

– the type of uncertainty is taken into account while issuing the BA (Step 21);

– the universality of solving the task of analyzing the state of BA objects due to the hierarchical nature of their description (expressions (3.1)–(3.10));

– the possibility of quick search for solutions due to the simultaneous search for a solution by several individuals (Steps 1–20);

– the adequacy of the obtained results (expressions (3.1)–(3.10));

– the ability to avoid the local extremum problem (Steps 1–20);

– the possibility of in-depth learning of knowledge bases of BA (Step 21).

The disadvantages of the proposed method include:

– the loss of informativeness while assessing the state of the analysis object due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of the analysis object;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the state of the heterogeneous analysis object;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous analysis object;

– to reduce the use of computing resources of decision-making support systems.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

3.2 THE DEVELOPMENT OF A SOLUTION SEARCH METHOD USING THE POPULATION ALGORITHM OF GLOBAL SEARCH OPTIMIZATION

The method of finding solutions using the population algorithm of global search optimization consists of the following sequence of steps:

Step 1. Input of output data. At this stage, the initial data available about the object to be analyzed are entered.

Step 2. Processing of raw data taking into account the degree of uncertainty.

Weed agents are placed taking into account the type of uncertainty about the state of the analysis object. At this stage, the type of uncertainty is taken into account about the object to be analyzed and initialization of the basic state model of the object to be analyzed [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. This is done with the help of appropriate correction coefficients.

The decisive criterion during classification is the percentage of incorrectly classified records. The smaller the share of incorrectly classified records, relative to the total mass, the better the classifier works on the given input parameters. In addition, the practical application of the classifier is strongly influenced by the time required for the training of the classifier and the classification itself. Due to the fact that the base of values can contain records about the object consisting of a set of parameters, which incurs a large expenditure of machine time, there is a need to reduce part of the informative features. The algorithm of invasive weeds is used as a classification algorithm. The dimensionality reduction of the space of informative features is carried out with the help of an advanced genetic algorithm and the training of individuals is carried out using the method of training evolving artificial neural networks.

Step 3. Formation of the optimization vector.

The optimization vector is presented in the form of an array X_i and takes values from 1 to $\sum_{j=1}^{L} kO_j$, where L is the number of system input variables; k is the number of variables describing the state of the system; O is the number of states for the j-th variable. The number of iterations N and the maximum number of vectors that can be saved after each iteration S are set.

The parameters $n_{\min}$ and $n_{\max}$ are set, which correspond to the minimum and maximum value of the descendant vectors that the parent vector can create at each iteration. The distribution parameter is specified. An initial vector X^0 is generated and the root mean square error and error-based fitness function φ^0 are calculated for it.

For each vector X^s (s takes values from 1 to the current number of vectors), n^s is determined – the number of vectors that can generate this vector:

$$n^s = \frac{n_{\max} - n_{\min}}{\varphi^{best} - \varphi^{worst}} \varphi^s + \frac{\varphi^{best} n_{\min} - \varphi^{worst} n_{\max}}{\varphi^{best} - \varphi^{worst}}, \tag{3.11}$$

where φ^{best}, φ^{worst} are the best and worst values of the fitness function.

Step 4. Creation of descendant vectors.

For each vector X, n^s new vectors are created according to the rules:

$$X_i^{s,j} = X_i^s + u,\ j = \left(\overline{1, n^s}\right),$$

$$u \sim N\left(0, \delta_N\right) = \delta_N \sqrt{-2\ln(a)} \cos(b),\ \delta_N = \delta\left(\frac{N - N'}{N}\right), \tag{3.12}$$

where N' is the number of the current iteration; a, b are random numbers [0, 1]; X_i^s is the component of the vector X^s; u is the distribution function.

Step 5. An arrangement of vectors.

All vectors, including parent and offspring, are ordered by decreasing error. If the number of vectors exceeds S, the population is reduced to S. If the current iteration is less than N, then go to Step 3.

Step 6. The reduction of the dimensionality of the feature space.

At this stage, an improved genetic algorithm is used to reduce the feature space, which was developed in work [4].

For the genetic algorithm [4] of dimensionality reduction, the input data of the algorithm are the table of observations, the parameters of the algorithm and the array of chromosomes.

Step 7. Learning knowledge bases.

In this research, the learning method based on evolving artificial neural networks, developed in the research [2], is used for training knowledge bases.

The end of algorithm.

The proposed method of finding solutions using the population algorithm of global search optimization.

The proposed method was simulated in the MathCad 14 software environment (USA). The task to be solved during the simulation was to determine the route of the ships in the operational zones of the Black and Azov seas in the conditions of hybrid actions of the enemy.

The list of hybrid actions that can be used against the forces (troops) of the Naval Forces in the course of their performance of the tasks of protecting the economic activity of the state at sea in the conditions of hybrid actions is as follows:

– a closure of marine areas through which the recommended routes of movement of vessels pass;

– the violation of norms of international maritime law;

– radio-electronic suppression of control and communication systems of forces (troops) of the Navy;

– cybernetic influence on the command and communication system of the forces (troops) of the Navy;

– the sabotage actions against the forces (troops) of the Navy and objects of economic activity of the state;

– special actions against the forces (troops) of the Navy and objects of economic activity of the state;

– informational and psychological influence on the personnel of the forces (troops) of the Navy.

The movement of ships outside the limits of the recommended routes requires the implementation of measures for their protection by the forces (troops) of the Navy through:

– the probability of the presence of sea mines on such routes;

– the possibility of negative impact on commercial shipping, which is carried out on new routes.

The negative impact is:

– the use of underwater diversionary forces and devices by the enemy against ships from the training areas that are close to the new shipping routes;

– the possibility of covert placement of sea mines by the enemy on new shipping routes, etc.

Then the task of choosing the movement of ships should be carried out using the forces and devices of the Navy of the Armed Forces of Ukraine and is reduced to the typical task of a traveling salesman, which consists in finding the most profitable route.

An example of the temporary closure for navigation of the maritime areas of the Black Sea under the pretext of naval exercises as of 21.09.2021, during which the determination of alternative routes for the movement of vessels is shown in **Fig. 3.1**.

The number of iterations, the minimum and maximum population size, the minimum and maximum number of scattered seeds, as well as the initial and final standard deviation are given in the **Table 3.5**. To simplify and speed up the numerical experiments, optimization was carried out for the two-dimensional functions of Himmelblau and Rosenbrock (**Table 3.5**).

Table 3.6 shows a comparison of the effectiveness of swarm optimization algorithms for solving the task of laying out the route of the movement of ships according to the Rastrigin function.

An analysis of the results given in **Table 3.6** allow to conclude that the improved algorithm of invasive weeds increases the efficiency of data processing at the level of 21–27 %.

The effectiveness of the proposed method, unlike the existing ones, is explained by the presence of additional procedures and the improvement of canonical ones.

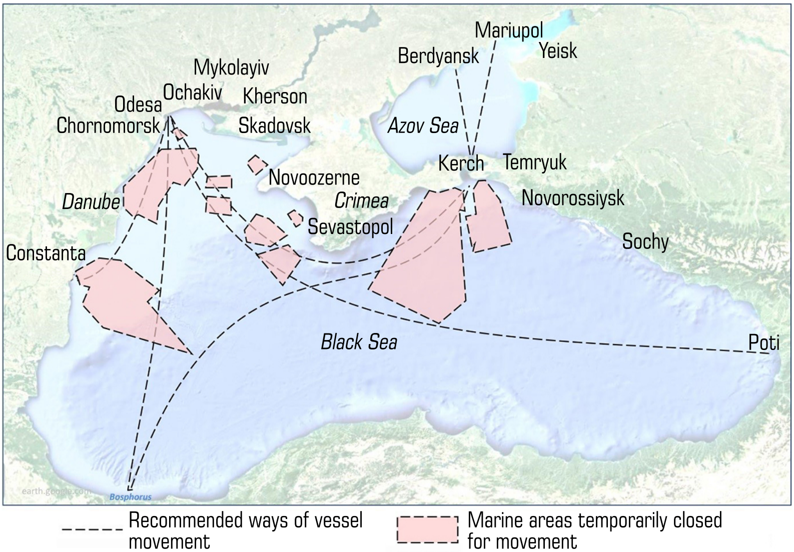

Fig. 3.1 An example of temporary closure for navigation of maritime areas of the Black Sea under the pretext of naval exercises as of 21.09.2021

Table 3.5 Computational efficiency results for route finding for Himmelblau and Rosenbrock functions

Function name	The number of iterations	Calculation error	
		Canonical IWO algorithm	**Improved IWO algorithm**
Himmelblau function $f(x,y)=\left(x^2+y-11\right)^2+\left(x+y^2-7\right)^2$	100	$20.9\cdot10^{-11}$	180.6
	500	$4.09\cdot10^{-11}$	180.6
	1000	$1.9\cdot10^{-11}$	180.6
Rosenbrock function $f(x,y)=-a\cdot\left(y-x^2\right)^2-\left(1-x^2\right)^2$	100	0.00746	–0.0019
	500	$12\cdot10^{-11}$	–0.000000001
	1000	$4.45\cdot10^{-11}$	–0.0000000006

The advantages of the proposed method are due to the following:

– while exhibiting weed agents, the type of uncertainty is taken into account (Step 2);

– the universality of solving the task of analyzing the state of objects of weed agents due to the hierarchical nature of their description (expressions (3.11)–(3.13));

– the possibility of quick search for solutions due to the simultaneous search for a solution by several individuals (Steps 1–7);

– the adequacy of the obtained results (expressions (3.11)–(3.13));
– the ability to avoid the local extremum problem (Steps 1–7);
– the possibility of in-depth learning of knowledge bases (Step 7);

Table 3.6 Comparison of the efficiency of swarm optimization algorithms for solving the task of laying the route of the movement of ships according to the Rastrigin function

The name of the optimization algorithm	Rastrigin function		
	10 parameters	50 parameters	100 parameters
Canonical algorithm of invasive weeds [23]	1	1	0.33519
Improved invasive weed algorithm	1	1	0.5
Canonical algorithm of ant colonies [16]	0.37	0.27	0.18
The canonical cuckoo algorithm [16]	0.96	0.7	0.29
Canonical firefly algorithm [19]	0.62	0.5	0.19
Canonical algorithm of bats [16, 22]	0.43	0.96	1
Canonical algorithm of artificial bee colonies	0.81	0.49	0.23
Canonical algorithm for finding a school of fish [22]	0.48	0.38	0.11
Canonical particle swarm algorithm [16, 22]	0.21	0.12	0.06
Random search [20]	0.18	0.14	0.07
Canonical algorithm of a flock of gray wolves [16, 22]	0	0	0

The main advantages of the proposed method are:

– it has a flexible hierarchical structure of indicators, which allows to reduce the task of multi-criteria evaluation of alternatives to one criterion or using a vector of indicators for selection;
– the unambiguousness of the obtained assessment of the state of the analysis object;
– the universality of application due to adaptation of the system of indicators during work;
– it does not accumulate learning error due to the use of the learning procedure;
– the possibility of comprehensive learning of the architecture and parameters of artificial neural networks;
– taking into account the type of uncertainty of the initial data while building models of a heterogeneous analysis object;
– the possibility of finding a solution in several directions;
– high reliability of the obtained solutions while searching for a solution in several directions;
– the reduction of the space of features while assessing the state of the analysis object using an improved genetic algorithm;
– the absence of falling into the local optimum trap.

The disadvantages of the proposed method include:

– the loss of informativeness while assessing the state of the analysis object due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of the analysis object;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the state of the heterogeneous analysis object;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous analysis object;

– to reduce the use of computing resources of decision-making support systems.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proving information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

3.3 THE DEVELOPMENT OF A SOLUTION SEARCH METHOD USING AN IMPROVED FISH SHOALING ALGORITHM

The proposed algorithm is an improved algorithm for schools of fish and consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available about the object to be analyzed are entered. The existing model of the analysis object is also initialized. At this stage, the solution matrix D is filled: each column is filled with a subset of F_k.

Step 2. Processing of raw data taking into account the degree of uncertainty.

At this stage, the type of uncertainty is taken into account about the object to be analyzed and initialization of the basic state model of the object to be analyzed [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. This is done with the help of correction coefficients.

Step 3. Individual stage of swimming agents. For each agent, the search for the best solution is performed in the neighborhood of the given solution. Since it is impossible to define the concept of a solution neighborhood for this type of problem, let's consider this procedure. While choosing a feeding source, the degree of noise about the condition of the object of evaluation is taken into account. The degree of noise is defined as total noise, partial noise and reliability.

Step 3.1. At the first stage, each column of the decision matrix D is considered. In each column, the elements d_{ik} and d_{ik+1} are interchanged, where i is the column number, k is an odd number.

Step 3.2. The second stage is similar to the first, with the difference that k is an even number.

Step 4. Checking the fitness function of the found solution. If the fitness function has improved, it is possible to consider that displacement has occurred. If the FA went outside the aquarium, it is possible to consider that there was no movement.

Step 5. FA feeding procedure. At this stage, it is necessary to fix the improvement of the current FA solution with respect to other FA in the jamb using the genetic algorithm developed in the work [22]. For this, such a property of FA as its weight will be used: $W = \Delta f_i / \max \Delta f_i$, where Δf_i is the value of the improvement of the value of the fitness function for another FA.

Step 6. Instinctive collective swimming FA. Each FA is affected by the entire population as a whole and the influence of an individual FA is proportional to its success in the individual swimming stage. After that, the entire population shifts by the size of the migration step, which characterizes the movement of the entire FA jamb.

To calculate the migration step, it is necessary to consider the process of moving one FA in the direction of another [24–30]. Let the current solution for this FA be D_1. In order to make a move to an BA that has a solution D_0, it is necessary to sequentially review the columns of the matrix D_0 from left to right. In each column, it is necessary to find out the relative location of the elements d_{ik} and d_{ik+1}, where i is the number of the column, k is an odd number. If in the matrix D_1 the location of the d_{ik} and d_{ik+1} elements does not coincide with the position of D_0, then the places of these elements are exchanged. If one of the elements is zero, then the replacement of elements in D_1 will be carried out if the modulus of the difference between the number of zeros over a nonzero element in the matrix D_0 and the number of zeros over a nonzero element in the matrix D_1 becomes smaller [31–36]. A move with a given probability α is a move in which the comparison and interchange of the elements d_{ik} and d_{ik+1} in matrices D_0 and D_1 is carried out with probability α.

To calculate the migration step, it is necessary for each fish to move with a given probability α. For this, the amount is calculated, $Sum = \sum_1^N \Delta f_i$ is the sum of improvements in the fitness function for each FA in the population. Then each FA in the jamb will move to other FA with probability $\alpha = \Delta f_i / Sum$.

Step 7. Calculation of the center of gravity of the jamb. This stage is preparatory for the next stage of the algorithm and consists in calculating a weighted decision regarding the total weight of the entire jamb. In this case, the calculation of the barycenter is performed as follows: the agents are arranged as the weight decreases; starting with the FA, the weight of which is the smallest, moving with a given probability ρ to the heaviest FA. The barycenter is the result of such a movement. The probability ρ is a controlled parameter.

Step 8. Collective voluntary swimming FA. At this stage of the algorithm, it is determined how the weight of the shoal of fish has changed compared to the previous iteration. If it has increased, then the population has approached the region of the maximum of the function, therefore it is necessary to narrow the circle of its search, thereby revealing the intensification properties. And vice versa: if the weight of the jamb has decreased, then the agents are looking for a maximum in the wrong place, so it is necessary to change the direction of the trajectory and identify diversification properties.

For this stage, it is necessary to determine the distance between two FA. Determining the distance takes place in two stages.

Step 8.1. At the first stage, it is necessary to sequentially review the columns of matrix D from left to right. In each column, it is necessary to find out the relative location of the elements d_{ik} and d_{ik+1}, where i is the number of the column, k is an odd number. If in the matrix $D_{barycenter}$ is the matrix of the agent's decision corresponding to the previously calculated barycenter – the location of the elements d_{ik} and d_{ik+1} does not coincide, then the degree of difference increases by 1, otherwise it does not increase.

Step 8.2. The second stage is carried out when k is an even number. The degree of difference is the distance between the two agents.

Collective voluntary swimming occurs with the help of moving in the direction of the FA barycenter with probability:

$$\beta = collStep \cdot rand(0;1) \cdot \frac{dist}{\max(dist)},$$

where $dist$ is the distance between the current agent and the barycenter; max($dist$) is the maximum distance between the FA in the jamb and the barycenter; $collStep$ is the free displacement step; $rand$ (0, 1) is a random number from 0 to 1.

Step 9. Changing the FA swimming parameters. At this stage, the step of individual movement of each FA changes depending on the iteration number. This procedure is used to increase the efficiency of the algorithm and faster convergence.

Step 10. Learning FA knowledge bases.

In this research, the learning method based on evolving artificial neural networks developed in the research [2] is used to learn the knowledge bases of each FA.

The end of algorithm.

The proposed method of finding solutions using the improved algorithm of schools of fish. In order to evaluate the effectiveness of the developed method, its comparative evaluation was performed based on the results of research presented in works [3–6, 23, 24, 37].

Simulation of the solution search processing method was carried out in accordance with Steps 1–10. The proposed method was simulated in the MathCad 14 software environment (USA). The assessment of the elements of the operational situation of the group of troops (forces) was the task to be solved during the simulation.

Initial data for assessing the state of the operational situation using the improved method:

– the number of sources of information about the state of the monitoring object is 3 (radio monitoring tools, remote sensing of the earth and unmanned aerial vehicles) To simplify the modeling, the same number of each tool was taken – 4 tools each;

– the number of informational signs by which the state of the monitoring object is determined – 12. Such parameters include: ownership, type of organizational and staff formation, priority, minimum width along the front, maximum width along the front. The number of personnel,

minimum depth along the flank, maximum depth along the flank, the number of samples of weapons and military equipment (WME), the number of types of WME samples and the number of communication devices), the type of operational construction are also taken into account;

– the variants of organizational and personnel formations are company, battalion, brigade.

The results of the comparative evaluation according to the criterion of efficiency of evaluation with known scientific studies are shown in **Table 3.7**.

Table 3.7 The results of problem solving

No. iteration	Method of branches and boundaries [17]	Genetic algorithm [12]	Canonical fish algorithm [23]	Improved fish algorithm
N	***T*, s**	***T*, s**	***T*, s**	***T*, s**
5	1.125	1.125	1.125	1.05
10	0.625	0.625	0.625	0.450
15	48.97	58.20	58.28	55.71
20	106.72	44.29	43.75	41.33
30	–0.1790	–0.0018	–0.0002	–0.00008
40	–0.158	–0.070	–0.069	–0.08
50	97.76	–974.30	–3.72	–331.18
100	–133.28	–195.71	–196.24	–198.12
200	7980.89	7207.49	7198.43	7022.85

As it can be seen from the **Table 3.7**, the gain of the specified method of finding solutions is from 11 to 15 % according to the criterion of data processing efficiency.

The advantages of the proposed method are due to the following:

– the type of uncertainty is taken into account while issuing FA (Step 2);

– the universality of solving the task of analyzing the state of FA objects due to the hierarchical nature of their description (Steps 1–10);

– the possibility of quick search for solutions due to the simultaneous search for a solution by several individuals (Steps 1–10, **Table 3.7**);

– the adequacy of the obtained results (Steps 1–10);

– the ability to avoid the local extremum problem (Steps 1–10);

– the possibility of in-depth learning of FA knowledge bases (Step 10).

The disadvantages of the proposed method include:

– the loss of informativeness while assessing the state of the analysis object due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of the analysis object;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the state of the heterogeneous analysis object;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous analysis object;

– to reduce the use of computing resources of decision-making support systems.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proving information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

3.4 THE DEVELOPMENT OF A SOLUTION SEARCH METHOD USING AN IMPROVED JUMPING FROG ALGORITHM

The method of finding solutions using the improved jumping frog algorithm consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available about the object to be analyzed are entered. The existing model of the analysis object is also initialized. The initialization of the initial population of frog agents (FA) is represented as a set of points of the space of permutations S_n with a Kendall metric of the form $\mathbf{S}=\left(s_1,\ s_2,\ \ldots,\ s_n\right)$. Every element of the vector corresponds to the features of the observation table, n is the number of features. The value of the element of the vector $s_i=0$ shows that the i-th feature does not participate in the classification $s_i=1$, means that the first feature is used by the classifier. In this step, a modification of algebraic operations is proposed, which allows the algorithm to operate on binary input vectors, since all arithmetic operators are replaced by logical ones. According to this idea, the operations of multiplication, addition and subtraction are replaced by conjunction, disjunction and strict disjunction, respectively:

$$S^{*}=r\wedge\left(S_b\oplus S_w\right)\vee S_w, \quad (3.13)$$

where r is an arbitrary binary vector.

Step 2. Processing of raw data taking into account the degree of uncertainty.

At this stage, the type of uncertainty is taken into account about the object to be analyzed and initialization of the basic state model of the object to be analyzed [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. This is done with the help of correction coefficients.

Step 3. Calculation of the value of the criterion of optimality of each permutation from the initial population of FA. This procedure is carried out using an improved genetic algorithm developed by the authors in the work [22].

Step 4. Ordering of the decision in order of decreasing value of the criterion of optimality.

Step 5. Global search of FA.

The procedure consists in sorting the input vectors by the value of the fitness function and dividing the population of FA into subgroups. Within each subgroup, a local search is carried out independently, in which the worst vectors are updated.

In the fitness function, it is important to take into account both the quality of the construction of the classification model and the share of selected features in order to reduce their number:

$$Fit(\mathbf{S}) = \alpha \times Error(\mathbf{S}) + (1-\alpha) \times \frac{n^*}{n}, \tag{3.14}$$

where $Error(\mathbf{S})$ is the classification error on the vector S; n^* is the number of elements in the vector $\mathbf{S}$ equal to one; α is the priority coefficient of one part of the function over another.

Vectors are updated according to the following principle. $\mathbf{Z}_b$ and $\mathbf{Z}_w$ are selected and are vectors with the best and worst value of the fitness function in the group. Then, the intermediate vector $\mathbf{Z}^*$ is calculated:

$$\mathbf{Z}^* = r \times c \times (\mathbf{Z}_b - \mathbf{Z}_w) + \mathbf{Z}_w, \tag{3.15}$$

where r is a uniformly distributed random number from 0 to 1; c is the vector update factor. If the value of the fitness function of the vector $\mathbf{Z}^*$ exceeds the fitness function of the vector $\mathbf{Z}_w$, $\mathbf{Z}^*$ replaces $\mathbf{Z}_w$. Otherwise, the $\mathbf{Z}^*$ vector is recomputed, but the global best vector $\mathbf{Z}_0$ is used instead of $\mathbf{Z}_b$. If even in this case it is not possible to improve the vector $\mathbf{Z}_w$, it is overwritten with a randomly generated deviation.

The algorithm accepts the following parameters as input: the number of subgroups G, the number of vectors in this group F, the update factor of input vectors c, the number of iterations for global and local search T_{gl} and T_k, respectively.

Separation of FA solutions into memplex blocks in such a way that the first FA in the sorted list goes to the first memplex, the second FA is entered into the second memplex, etc. This continues until all FA are distributed into the specified number of memplexes.

Step 6. Regulation of the speed of movement of the FA.

In each memplex with number $k \in \{1, 2, \ldots, K\}$ find the best s_{k1} and the worst s_{k2} solution. To perform this step, it is necessary to have some continuous vector that characterizes the features.

Most often, such a vector is a velocity vector. There is no such vector in the canonical FA algorithm, so it is suggested to calculate the speed of the worst vector $\mathbf{S}_w$ as follows:

$$\mathbf{V} = \left(\mathbf{S}_0 - \mathbf{S}_w\right) \times \mathbf{r}_1 + \left(\mathbf{S}_b - \mathbf{S}_w\right) \times \mathbf{r}_2, \tag{3.16}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are vectors filled with random real values in the range from 0 to 1. Then, the received value must be matched with a binary equivalent.

Step 6.1. At the first step, the transformation function, taking as input the value of the velocity of the element of the vector $\mathbf{S}_w$, calculates a number that belongs to the range [0; 1].

Step 6.2. Direct update of elements according to the transformation rule.

There are several families of transformation functions. The two most commonly used families are S- and V-shaped charts. As an S-shaped transformation function, the basic version of the sigmoid was used:

$$F_1\left(v_i\right) = 1 \big/ \left(1 + e^{-v_i}\right), \tag{3.17}$$

where v_i is the speed value of the *i*-th element.

V-shaped functions have a large variety of variations, so two functions were chosen for the research. The first is calculated using the hyperbolic tangent:

$$F_2\left(v_i\right) = \left|tanh\left(v_i\right)\right|, \tag{3.18}$$

the second is given by the following expression:

$$F_3\left(v_i\right) = \left|v_i \big/ \sqrt{1 + v_i^2}\right|. \tag{3.19}$$

Transformational rules differ in the principle of updating elements. In the first rule an element R_1 is given a strictly binary value:

$$\text{if } rand < F\left(v_i\right), \text{ then } s_i = 1, \text{ otherwise } s_i = 0, \tag{3.20}$$

where $F\left(v_j\right)$ is one of the three transformation functions; *rand* is a uniformly distributed random number; $rand \in \left[0;\ 1\right]$. The second rule R_2 either replaces the element with its opposite or does not change it:

$$\text{if } rand < F\left(v_i\right), \text{ then } s_i = s_i \oplus 1. \tag{3.21}$$

Step 7. Improvement of the position of FA in the search space. An improvement of the position of the worst FA by moving it in the direction of the best FA, taking into account the degree of noise of the original data [3]. This happens using the crossover operator $s = Cross(s_{k1},\ s_{k2})$.

Step 8. An improvement of working conditions of FA.

If the previous operation does not improve the solution, try to improve the position of the worst FA by moving it in the direction of the globally best FA $s=Cross(s_{k1}, s_{11})$.

Step 9. Rearrangement of FA.

If the last operation does not lead to an improvement in the position of the FA, then instead of the FA, randomly create a new FA in the search area – a permutation.

Step 10. FA unification of all memplexes into one group.

The function compares two binary vectors element by element; if the value of the element at the same position coincides, then the given value will be written to this position in the resulting vector. Otherwise, a random number is generated from the interval from 0 to 1 [24–30]. If it is less than or equal to 0.5, then the corresponding position of the new vector is written by the element from the worse vector. Otherwise, an element from a better vector will be displayed at this location.

Thus, the merge function can be given as follows:

$$merge(S_w, S_b) = \begin{cases} s_i^* = s_{wi} = s_{bi}, \text{ if } s_{wi} = s_{bi}; \\ s_i^* = s_{wi}, \text{ if } s_{wi} \neq s_{bi} \text{ and } rand \leq 0.5; \\ s_i^* = s_{bi}, \text{ if } s_{wi} \neq s_{bi} \text{ and } rand > 0.5, \end{cases} \quad (3.22)$$

where *rand* is a random, uniformly distributed number, $rand \in [0;1]$.

Step 11. If the conditions for completing the algorithm are not met, then proceed to Step 3.

Step 12. Search for the best FA.

The last globally best FA corresponds to a suboptimal solution to the problem.

Step 13. Training of the knowledge bases of FA.

In the research, the learning method based on evolving artificial neural networks, developed in the research [2], is used to learn the knowledge bases of each FA.

The end of algorithm.

The proposed method of searching for solutions using the improved FA.

Simulation of the work of the solution search method was carried out in accordance with Steps 1–13. The proposed method was simulated in the MathCad 14 software environment (USA). The assessment of elements of the operational situation of the group of troops (forces) was the task to be solved during the simulation.

Initial data for assessing the state of the operational situation using the improved method:

– the number of sources of information about the state of the monitoring object is 3 (radio monitoring tools, remote sensing of the earth and unmanned aerial vehicles) To simplify the modeling, the same number of each tool was taken – 4 tools each;

– the number of informational signs by which the state of the monitoring object is determined – 12. Such parameters include: ownership, type of organizational and staff formation, priority, minimum width along the front, maximum width along the front. The number of personnel, minimum depth along the flank, maximum depth along the flank, the number of samples of weapons

and military equipment (WME), the number of types of WME samples and the number of communication devices), type of operational construction are also taken into account;

– the variants of organizational and personnel formations are company, battalion, brigade.

Two-time cross-validation was used in the simulation. The algorithm was run 30 times on each sample. The resulting classification quality scores were averaged over the number of runs and two samples of each signature. The following indicators were recorded: overall accuracy, type I error (False Rejection Rate, FRR), type II error (False Acceptance Rate, FAR) and the number of features remaining in the set.

The structure of the classifier was generated by the algorithm taking into account the extreme values of the classes. Triangular functions of accessories were chosen as terms. The FA parameters are as follows: the number of groups is 4, each of them has 10 vectors; local iterations 5, global iterations 50. The coefficient in the fitness function was equal to 0.5. **Table 3.8** presents the list of binarization methods and their designations used in FA.

Table 3.8 Numbering of FA binarization

Method	
MAO	Modified algebraic operations (3)
Merge	Merge operation (4)
MAOm	Modified algebraic operations + merger operation (5)
F1R1	Transformation function F_1(7) + rule R_1(10)
F2R1	Transformation function F_2(8) + rule R_1(10)
F3R1	Transformation function F_3(9) + rule R_1(10)
F1R2	Transformation function F_1(7) + rule R_2(11)
F2R2	Transformation function F_2(8) + rule R_2(11)
F3R2	Transformation function F_3(9) + rule R_2(11)
F1R1m	Transformation function F_1(7) + rule R_1(10) + merge operation
F2R1m	Transformation function F_2(8) + rule R_1(10) + merge operation
F3R1m	Transformation function F_3(9) + rule R_1(10) + merge operation
F1R2m	Transformation function F_1(7) + rule R_2(11) + merge operation
F2R2m	Transformation function F_2(8) + rule R_2(11) + merge operation
F3R2m	Transformation function F_3(9) + rule R_2(11) + merge operation

Table 3.9 contains the obtained classification indicators on test samples. The best results by indicator are marked in bold.

The average ranks obtained while comparing the results of all users using the Friedman criterion are presented in **Table 3.10**.

Table 3.9 The results of building a fuzzy classifier on subsets of features selected by the FA

Method	Precision	FRR	FAR	Signs
MAO	83.9±7.5	25.1±14.0	7.1±8.1	43.9±4.3
merge	86.7±6.9	20.2±13.0	6.3±7.6	34.2±3.6
MAOm	85.5±7.7	21.1±14.3	7.9±9.2	20.5±4.1
F1R1	85.2±7.4	23.0±13.3	6.5±8.0	40.4±4.1
F2R1	85.6±7.3	21.9±13.5	7.0±8.2	28.0±4.2
F3R1	86.1±7.2	20.7±13.7	7.0±8.3	24.6±4.3
F1R2	83.8±7.6	25.4±14.2	7.1±8.2	43.5±4.1
F2R2	87.0±7.2	19.3±13.3	6.8±8.1	30.0±4.0
F3R2	**87.1**±7.0	19.4±13.1	6.4±7.7	31.1±4.0
F1R1m	85.3±7.3	23.2±13.5	**6.2**±7.8	39.1±3.9
F2R1m	85.9±7.5	20.1±13.9	8.0±9.3	19.7±5.1
F3R1m	85.6±7.7	20.1±14.1	8.6±9.6	**17.6**±4.6
F1R2m	84.2±7.4	24.2±13.6	7.5±8.6	44.8±4.1
F2R2m	**87.1**±7.0	**19.0**±12.9	6.9±8.1	29.4±4.0
F3R2m	86.7±6.9	20.0±13.0	6.7±7.9	30.1±3.9

Table 3.10 Average ranks according to the Friedman criterion

Method	Precision	FRR	FAR	Signs
MAO	4.8	11.3	7.8	13.9
merge	10.9	6.2	6.1	10.0
MAOm	5.9	8.9	10.5	2.8
F1R1	8.1	9.2	6.4	11.9
F2R1	7.1	9.0	8.4	5.3
F3R1	8.0	8.0	8.1	4.0
F1R2	4.7	12.1	7.5	13.5
F2R2	11.0	4.6	7.3	7.4
F3R2	11.6	5.1	5.7	8.9
F1R1m	8.4	9.7	5.3	11.1
F2R1m	6.9	7.4	11.0	2.2
F3R1m	6.1	7.6	12.0	1.0
F1R2m	5.5	10.2	8.9	14.6
F2R2m	11.1	4.6	8.0	6.0
F3R2m	10.1	6.2	7.1	7.5

The best overall accuracy results were demonstrated by V-shaped transformation functions with a rule R_2(23). At the same time, the function F_2(21) showed the best result in combination with the fusion operation and the function F_3(22) – individually. The error of the first kind turned out to be the smallest during use F_2 with the rule R_2 both in the absence and in the presence of a merger operation. The smallest error of the second kind was obtained using F_1(20) and rules R_1 (22) in combination with the merging operation. The function F_3 showed the best feature reduction ability with the rule R_1 and the merger operation.

The results of the comparative evaluation according to the criterion of efficiency of evaluation are shown in the **Table 3.11**.

Table 3.11 The results of solving the problem

No. iterations	Method of branches and boundaries [17]	Genetic algorithm [12]	Canonical FA [23]	Improved FA
N	***T*, s**	***T*, s**	***T*, s**	***T*, s**
5	1.125	1.125	1.125	1.05
10	0.625	0.625	0.625	0.450
15	48.97	58.20	58.28	55.71
20	106.72	44.29	43.75	41.33
30	–0.1790	–0.0018	–0.0002	–0.00008
40	–0.158	–0.070	–0.069	–0.08
50	97.76	–974.30	–3.72	–331.18
100	–133.28	–195.71	–196.24	–198.12
200	7980.89	7207.49	7198.43	7022.85

As can be seen from the **Table 3.11**, the gain of the specified solution search method is from 14 to 18 % according to the criterion of data processing efficiency.

The main advantages of the proposed method are:

– it has a flexible hierarchical structure of indicators, which allow to reduce the task of multi-criteria evaluation of alternatives to one criterion or using a vector of indicators for selection;

– a smaller error in assessing the state of the object due to the improved transformation procedure;

– an unambiguousness of the obtained assessment of the state of the analysis object;

– the universality of application due to adaptation of the system of indicators during work;

– it does not accumulate learning error due to the use of the learning procedure;

– the possibility of comprehensive learning of the architecture and parameters of artificial neural networks;

– taking into account the type of uncertainty of the initial data while building models of a heterogeneous object of analysis;

– the possibility of finding a solution in several directions;

– high reliability of the obtained solutions while searching for a solution in several directions;

– the absence of falling into the local optimum trap.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proving information from intelligence sources.

The advantages of the proposed method are due to the following:

– the type of uncertainty is taken into account while setting up FA (Step 2);

– during the movement of the FA, the degree of noise of the data on the object state is taken into account;

– the universality of solving the task of analyzing the state of FA objects due to the hierarchical nature of their description (Steps 1–13);

– the possibility of quick search for solutions due to the simultaneous search for a solution by several individuals (Steps 1–13);

– the adequacy of the obtained results (Steps 1–13);

– the ability to avoid the local extremum problem (Steps 1–13);

– taking into account the degree of noise of the data on the object state during the movement of the vehicle (Step 7);

– using improved binarization procedures (Step 1, expression (3.13));

– by calculating the speed of the vehicle (Step 6, expressions (3.16), (3.17));

– the possibility of in-depth learning of the knowledge bases of FA (Step 13).

The disadvantages of the proposed method include:

– the loss of informativeness while assessing the state of the analysis object due to the construction of the membership function;

– lower accuracy of assessment on a single parameter of assessment of the state of the analysis object;

– the loss of credibility of the obtained solutions while searching for a solution in several directions at the same time;

– lower assessment accuracy compared to other assessment methods.

This method will allow:

– to assess the state of the heterogeneous analysis object;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous analysis object;

– to reduce the use of computing resources of decision-making support systems.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

CONCLUSIONS

1. The method implementation algorithm is determined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of the BA search;

– to carry out the initial exhibition of BA individuals, taking into account the type of uncertainty;

– to carry out accurate training of BA individuals using expressions;

– to determine the best BA individuals using a genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

2. Example of using the proposed method is conducted on the example of assessing and forecasting the state of the operational situation of a group of forces. The specified example showed an increase in the efficiency of data processing at the level of 13–21 % due to the use of additional improved procedures of adding correction coefficients for uncertainty and noisy data, selection of BA and training of BA.

3. The method implementation algorithm is defined, which allows:

– to take into account the type of data uncertainty;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of search by weed agents;

– to conduct an initial display of individuals of weed agents, taking into account the type of uncertainty;

– to carry out accurate training of individuals of weed agents;

– to reduce the space of features while assessing the state of the analysis object with the help of an improved genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

4. An example of the use of the proposed method was carried out on the example of choosing the route of the movement of ships in the operational zones of the Black and Azov Seas. The specified example showed an increase in the efficiency of data processing at the level of 21–27 % due to the use of additional improved procedures for the selection of individuals and deep learning.

5. The method implementation algorithm is defined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of the FA search;

– to carry out the initial display of FA individuals taking into account the type of uncertainty;

– to carry out accurate training of FA individuals;

– to determine the best FA individuals using a genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

6. Example of using the proposed method is conducted on the example of assessing and forecasting the state of the operational situation of a group of forces. The specified example showed an increase in the efficiency of data processing at the level of 18–25 % due to the use of additional improved procedures of adding correction coefficients for uncertainty and noisy data, selection of FA, as well as training of FA.

7. The method implementation algorithm was determined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of the traffic of vehicles;

– to carry out the initial display of individuals of FA, taking into account the type of uncertainty;

– to carry out accurate training of FA individuals;

– to determine the best individuals of FA using a genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to calculate the speed of the vehicle;

– to avoid the problem of local extremum.

8. Example of using the proposed method is conducted on the example of assessing and forecasting the state of the operational situation of a group of forces. The specified example showed an increase in the efficiency of data processing at the level of 14–18 % due to the use of additional improved procedures for adding correction coefficients for uncertainty and noise in data, FA selection, calculation of FA movement speed and FA training.

REFERENCES

1. Shyshatskyi, A. V., Bashkyrov, O. M., Kostyna, O. M. (2015). Rozvytok intehrovanykh system zviazku ta peredachi danykh dlia potreb Zbroinykh Syl. Ozbroiennia ta viiskova tekhnika, 1 (5), 35–40.
2. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskyi, R., Repilo, I. et al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 37–47. https://doi.org/10.15587/1729-4061.2020.203301
3. Sova, O., Shyshatskyi, A., Salnikova, O., Zhuk, O., Trotsko, O., Hrokholskyi, Y. (2021). Development of a method for assessment and forecasting of the radio electronic environment. EUREKA: Physics and Engineering, 4, 30–40. https://doi.org/10.21303/2461-4262.2021.001940
4. Pievtsov, H., Turinskyi, O., Zhyvotovskyi, R., Sova, O., Zvieriev, O., Lanetskii, B., Shyshatskyi, A. (2020). Development of an advanced method of finding solutions for neuro-fuzzy expert systems of analysis of the radioelectronic situation. EUREKA: Physics and Engineering, 4, 78–89. https://doi.org/10.21303/2461-4262.2020.001353

5. Zuiev, P., Zhyvotovskyi, R., Zvieriev, O., Hatsenko, S., Kuprii, V., Nakonechnyi, O. et al. (2020). Development of complex methodology of processing heterogeneous data in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 14–23. https://doi.org/10.15587/1729-4061.2020.208554
6. Shyshatskyi, A., Zvieriev, O., Salnikova, O., Demchenko, Ye., Trotsko, O., Neroznak, Ye. (2020). Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering, 9 (4), 5583–5590. https://doi.org/10.30534/ijatcse/2020/206942020
7. Nechyporuk, O., Sova, O., Shyshatskyi, A., Kravchenko, S., Nalapko, O., Shknai, O. et al. (2023). Development of a method of complex analysis and multidimensional forecasting of the state of intelligence objects. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 31–41. https://doi.org/10.15587/1729-4061.2023.276168
8. Koval, V., Nechyporuk, O., Shyshatskyi, A., Nalapko, O., Shknai, O., Zhyvylo, Y. et al. (2023). Improvement of the optimization method based on the cat pack algorithm. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 41–48. https://doi.org/10.15587/1729-4061.2023.273786
9. Ko, Y.-C., Fujita, H. (2019). An evidential analytics for buried information in big data samples: Case study of semiconductor manufacturing. Information Sciences, 486, 190–203. https://doi.org/10.1016/j.ins.2019.01.079
10. Ramaji, I. J., Memari, A. M. (2018). Interpretation of structural analytical models from the coordination view in building information models. Automation in Construction, 90, 117–133. https://doi.org/10.1016/j.autcon.2018.02.025
11. Pérez-González, C. J., Colebrook, M., Roda-García, J. L., Rosa-Remedios, C. B. (2019). Developing a data analytics platform to support decision making in emergency and security management. Expert Systems with Applications, 120, 167–184. https://doi.org/10.1016/j.eswa.2018.11.023
12. Chen, H. (2018). Evaluation of Personalized Service Level for Library Information Management Based on Fuzzy Analytic Hierarchy Process. Procedia Computer Science, 131, 952–958. https://doi.org/10.1016/j.procs.2018.04.233
13. Chan, H. K., Sun, X., Chung, S.-H. (2019). When should fuzzy analytic hierarchy process be used instead of analytic hierarchy process? Decision Support Systems, 125, 113114. https://doi.org/10.1016/j.dss.2019.113114
14. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. Future Generation Computer Systems, 91, 620–633. https://doi.org/10.1016/j.future.2018.06.046
15. Gödri, I., Kardos, C., Pfeiffer, A., Váncza, J. (2019). Data analytics-based decision support workflow for high-mix low-volume production systems. CIRP Annals, 68 (1), 471–474. https://doi.org/10.1016/j.cirp.2019.04.001
16. Harding, J. L. (2013). Data quality in the integration and analysis of data from multiple sources: some research challenges. The International Archives of the Photogrammetry,

CHAPTER 3

Remote Sensing and Spatial Information Sciences, XL-2/W1, 59–63. https://doi.org/10.5194/isprsarchives-xl-2-w1-59-2013

17. Kosko, B. (1986). Fuzzy cognitive maps. International Journal of Man-Machine Studies, 24 (1), 65–75. https://doi.org/10.1016/s0020-7373(86)80040-2
18. Orouskhani, M., Orouskhani, Y., Mansouri, M., Teshnehlab, M. (2013). A Novel Cat Swarm Optimization Algorithm for Unconstrained Optimization Problems. International Journal of Information Technology and Computer Science, 5(11), 32–41. https://doi.org/10.5815/ijitcs.2013.11.04
19. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. https://doi.org/10.15587/1729-4061.2019.180197
20. Mahdi, Q. A., Shyshatskyi, A., Prokopenko, Y., Ivakhnenko, T., Kupriyenko, D., Golian, V. et al. (2021). Development of estimation and forecasting method in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 51–62. https://doi.org/10.15587/1729-4061.2021.232718
21. Gorokhovatsky, V., Stiahlyk, N., Tsarevska, V. (2021). Combination method of accelerated metric data search in image classification problems. Advanced Information Systems, 5 (3), 5–12. https://doi.org/10.20998/2522-9052.2021.3.01
22. Levashenko, V., Liashenko, O., Kuchuk, H. (2020). Building Decision Support Systems based on Fuzzy Data. Advanced Information Systems, 4 (4), 48–56. https://doi.org/10.20998/2522-9052.2020.4.07
23. Meleshko, Y., Drieiev, O., Drieieva, H. (2020). Method of identification bot profiles based on neural networks in recommendation systems. Advanced Information Systems, 4 (2), 24–28. https://doi.org/10.20998/2522-9052.2020.2.05
24. Kuchuk, N., Merlak, V., Skorodelov, V. (2020). A method of reducing access time to poorly structured data. Advanced Information Systems, 4 (1), 97–102. https://doi.org/10.20998/2522-9052.2020.1.14
25. Shyshatskyi, A., Tiurnikov, M., Suhak, S., Bondar, O., Melnyk, A., Bokhno, T., Lyashenko, A. (2020). Method of Assessment of the Efficiency of the Communication of Operational Troop Grouping System. Advanced Information Systems, 4 (1), 107–112. https://doi.org/10.20998/2522-9052.2020.1.16
26. Raskin, L., Sira, O. (2016). Method of solving fuzzy problems of mathematical programming. Eastern-European Journal of Enterprise Technologies, 5 (4 (83)), 23–28. https://doi.org/10.15587/1729-4061.2016.81292
27. Lytvyn, V., Vysotska, V., Pukach, P., Brodyak, O., Ugryn, D. (2017). Development of a method for determining the keywords in the slavic language texts based on the technology of web mining. Eastern-European Journal of Enterprise Technologies, 2 (2 (86)), 14–23. https://doi.org/10.15587/1729-4061.2017.98750

28. Stepanenko, A., Oliinyk, A., Deineha, L., Zaiko, T. (2018). Development of the method for decomposition of superpositions of unknown pulsed signals using the secondorder adaptive spectral analysis. Eastern-European Journal of Enterprise Technologies, 2 (9 (92)), 48–54. https://doi.org/10.15587/1729-4061.2018.126578
29. Tarkhan, A. B., Zhuravskyi, Y., Shyshatskyi, A., Pluhina, T., Dudnyk, V., Kiris, I. et al. (2023). Development of a solution search method using an improved fish school algorithm. Eastern-European Journal of Enterprise Technologies, 4 (4 (124)), 27–33. https://doi.org/10.15587/1729-4061.2023.284315
30. Koval, M., Sova, O., Orlov, O., Shyshatskyi, A., Artabaiev, Y., Shknai, O. et al. (2022). Improvement of complex resource management of special-purpose communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 34–44. https://doi.org/10.15587/1729-4061.2022.266009
31. Sova, O., Radzivilov, H., Shyshatskyi, A., Shvets, P., Tkachenko, V., Nevhad, S. et al. (2022). Development of a method to improve the reliability of assessing the condition of the monitoring object in special-purpose information systems. Eastern-European Journal of Enterprise Technologies, 2 (3 (116)), 6–14. https://doi.org/10.15587/1729-4061.2022.254122
32. Mohammed, B. A., Zhuk, O., Vozniak, R., Borysov, I., Petrozhalko, V., Davydov, I. et al. (2023). Improvement of the solution search method based on the cuckoo algorithm. Eastern-European Journal of Enterprise Technologies, 2 (4 (122)), 23–30. https://doi.org/10.15587/1729-4061.2023.277608
33. Mamoori, G. A., Sova, O., Zhuk, O., Repilo, I., Melnyk, B., Sus, S. et (2023). The development of solution search method using improved jumping frog algorithm. Eastern-European Journal of Enterprise Technologies, 4 (3 (124)), 45–53. https://doi.org/10.15587/1729-4061.2023.285292
34. Shyshatskyi, A., Romanov, O., Shknai, O., Babenko, V., Koshlan, O., Pluhina, T. et al. (2023). Development of a solution search method using the improved emperor penguin algorithm. Eastern-European Journal of Enterprise Technologies, 6 (4 (126)), 6–13. https://doi.org/10.15587/1729-4061.2023.291008
35. Thamer, K. A., Sova, O., Shaposhnikova, O., Yashchenok, V., Stanovska, I., Shostak, S. et al. (2024). Development of a solution search method using a combined bio-inspired algorithm. Eastern-European Journal of Enterprise Technologies, 1 (4 (127)), 6–13. https://doi.org/10.15587/1729-4061.2024.298205
36. Shyshatskyi, A. V., Zhuk, O. V., Neronov, S.M., Protas, N. M., Kashkevych S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam metaevrystychnykh alhorytmiv. C91 Moderní aspekty vědy: XL. Mezinárodní Ekonomický Institut s.r.o., 529–557. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf
37. Shyshatskyi, A. V., Matsyi, O. B., Yashchenok, V. Zh., Trotsko, O. O., Kashkevych, S. O. (2024). Sukupnist metodyk pidvyshchennia operatyvnosti pryiniattia rishen z vykorystanniam kombinovanykh metaevrystychnykh alhorytmiv. C91 Moderní aspekty vědy: XL. Mezinárodní

Ekonomický Institut s.r.o., 558–594. Available at: http://perspectives.pp.ua/public/site/mono/mono-40.pdf

38. Kashkevych, S. O. (2023). Analiz modelei doslidzhennia skladnykh tekhnichnykh system. Modern scientific technologies and solutions of scientists to create the latest ideas. London, 290–294. Avaialble at: https://isg-konf.com/uk/modern-scientific-technologies-and-solutions-of-scientists-to-create-the-latest-ideas/
39. Kashkevych, S. O., Voznytsia, A. S. (2023). The development of methods for finding solutions using the improved of locusts swarm algorithm. Global problems of improving scientific inventions. Kopenhahen, 271–276. Avaialble at: https://isg-konf.com/uk/global-problems-of-improving-scientific-inventions/
40. Shyshatskyi, A. V., Lytvynenko, O. I., Zhuk, O. V., Artiukh, S. H., Kashkevych, S. O. (2023). Rozrobka metodyky pidvyshchennia operatyvnosti pryiniattia rishen v orhanizatsiino-tekhnichnykh systemakh. Development trends and improvement of old methods. Varshava, 422–431. Avaialble at: https://isg-konf.com/uk/development-trends-and-improvement-of-old-methods/

DOI: 10.15587/978-617-8360-04-7.CH4

Svitlana Kashkevich, Heorhii Kuchuk, Ganna Plekhova,
Viacheslav Davydov, Oleksandr Yefymenko, Yurii Beketov

CHAPTER 4

THE DEVELOPMENT OF METHODS OF LEARNING ARTIFICIAL NEURAL NETWORKS OF INTELLIGENT DECISION-MAKING SUPPORT SYSTEMS

ABSTRACT

A set of training methods for artificial neural networks of intelligent decision-making support systems has been developed. A distinctive feature of the proposed methods is that not only the synaptic weights of the artificial neural network, but also the type and parameters of the membership function are trained. If it is impossible to ensure the specified quality of functioning of artificial neural networks due to learning the parameters of the artificial neural network, the learning of the architecture of artificial neural networks takes place. The choice of the architecture, type and parameters of the membership function takes place taking into account the computing resources of the tool and taking into account the type and amount of information entering the input of the artificial neural network. Due to the use of the proposed methods, there is no accumulation of learning errors of artificial neural networks as a result of processing information received at the input of artificial neural networks. Also, a distinctive feature of the developed methods is that no preliminary calculation data is required for data calculation. The development of the proposed methods is due to the need to train artificial neural networks for intelligent decision-making support systems, with the aim of processing a larger amount of information, with the unambiguity of the decisions being made.

According to the results of the research, it was established that the specified training methods provide an average of 10–18 % higher training efficiency of artificial neural networks and do not accumulate errors during training. These methods will make it possible to conduct training of artificial neural networks, to determine effective measures to increase the efficiency of the functioning of artificial neural networks.

The use of the specified methods will allow to increase the efficiency of the functioning of artificial neural networks due to the learning of the parameters and architecture of artificial neural networks, to reduce the use of computing resources of decision-making support systems.

The developed methods will make it possible to develop measures aimed at increasing the effectiveness of learning artificial neural networks; increase the efficiency of information processing in artificial neural networks.

KEYWORDS

Artificial neural networks, learning, responsiveness, information processing, intelligent decision-making support systems.

4.1 THE DEVELOPMENT OF A METHOD FOR LEARNING ARTIFICIAL NEURAL NETWORKS WITH AN EVOLVING ARCHITECTURE

Decision making support systems (DMSS) are actively used in all spheres of people's life. They are especially popular while processing large data sets, providing informational support for the decision-making process by decision makers.

The conducted analysis of scientific research [1–10] shows that currently the basis of existing DMSS are artificial intelligence methods.

The creation of intelligent DMSS became a natural continuation of the widespread use of classical DMSS. Intelligent DMSS provide information support for all production processes and services of enterprises (organizations, institutions). Designing, manufacturing and sales of products, financial and economic analysis, planning, personnel management, marketing, support for the creation (operation, repair) of products and prospective planning are carried out with the help of intelligent DMSS. Also, the mentioned intellectual DMSS have been widely used to solve specific military tasks, namely [1, 2]:

– planning deployment, operation of communication and data transmission systems;

– automation of the management of troops and weapons;

– collection, processing and generalization of intelligence information about the state of intelligence objects, etc.

The main tool for solving calculation and other tasks in modern intelligent DMSS are artificial neural networks (ANN), which are evolving.

Evolving ANN have both universal approximating properties and fuzzy inference capabilities. Evolving ANN have become widely used to solve various tasks of intelligent data analysis, identification, emulation, forecasting, intelligent management, etc.

Evolving ANN ensure stable operation in conditions of nonlinearity, uncertainty, stochasticity and chaos, various types of disturbances and disturbances.

Despite their rather successful application for solving a wide range of tasks of intelligent data analysis, these systems have a number of disadvantages associated with their use.

Among the most significant shortcomings, the following can be distinguished:

1. The complexity of choosing a system architecture. As a rule, a model based on the principles of computational intelligence has a fixed architecture. In the context of ANN, this means that a neural network has a fixed number of neurons and connections. In this regard, adaptation of the system to new data arriving for processing, which have a different nature from the previous data, may be problematic.

2. Training in batch mode and training during several epochs requires significant time resources. Such systems are not adapted to work in online mode with a fairly high rate of arrival of new data for processing.

3. Many of the existing systems of computational intelligence cannot determine the evolving rules by which the system develops and can also represent the results of their work in terms of natural language.

Thus, the task of developing new training methods for ANN that will allow to solve the mentioned difficulties is urgent.

An analysis of scientific works [1–17] showed that well-known learning methods are used to train artificial neural networks. These methods are usually focused on learning synaptic weights or membership functions. The use of well-known algorithms (methods, techniques) for learning artificial neural networks, even with improved characteristics, does not satisfy the existing and prospective requirements for them.

Taking into account the above, the aim is to develop methods for learning artificial neural networks for intelligent decision-making support systems to solve the following tasks:

– increasing the amount of information that can be processed by artificial neural networks;

– increasing the reliability of decision making by intelligent decision-making support systems;

– increasing the speed of adaptation of the architecture and parameters of artificial neural networks in accordance with emerging tasks;

– the prevention of deadlocks under our training of artificial neural networks;

– ensuring the predictability of the learning process of artificial neural networks;

– ensuring the unambiguity of decisions made by intelligent decision-making support systems.

The architecture of the evolving multi-layered neuro-fuzzy system is shown in **Fig. 4.1** consists of five consecutive layers.

The neuro-fuzzy system is fed to the input (zero) layer $(n \times 1)$ – measurement vector of image signals $x(k) = (x_1(k), x_2(k), \ldots, x_n(k))^{\mathrm{T}}$ (here $k = 1,2\ldots$ is the current discrete time) to be processed.

The first hidden layer contains nh membership functions $\mu_{li}(x_i)$, $i = 1,2,\ldots,n$; $l = 1,2,\ldots,h$. Thus, h membership functions are given for each input. The first hidden layer performs fuzzification of the input space. At the same time, it is meant that both the actual parameters of these functions and their number should be adjusted during the learning-evolution process: in **Fig. 4.1** is the number of nodes of the first hidden layer μ_h.

The second hidden layer provides aggregation of membership levels calculated in the first hidden layer and consists of h multiplication blocks.

The third hidden layer is the layer that adjusts the synaptic weights $w_1, w_2, \ldots, w_h$, which are to be determined in the process of supervised learning.

The fourth hidden layer is formed by two adders, which calculates the sums of the output signals of the second and third hidden layers.

And, finally, in the fifth (output) layer, defuzzification is performed, as a result of which the output signal NFS (neuro-fuzzy system) is calculated $\hat{y}(k)$.

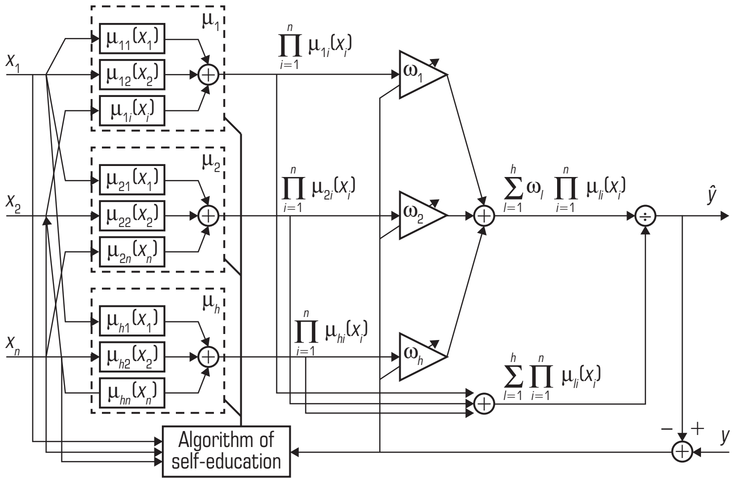

Fig. 4.1 The architecture of a multi-layered evolving neuro-fuzzy system

Thus, if a vector signal is applied to the input of the neuro phase of the system $x(k)$, the elements of the first hidden layer phase it by calculating the membership levels $0 < \mu_{li}(x_i(k)) \le 1$. Usually, traditional Gaussians are used as membership functions:

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2} \right), \tag{4.1}$$

where c_{li} is the center parameter of the l-th membership function of the i-th input; σ_i is the width parameter of the membership function of the i-th input.

It is worth noting that preliminary normalization of data for some interval, for example, $-1 \le x_i(k) \le 1$ allows to simplify calculations, because the parameters of the width can be assumed to be the same for all inputs, i.e. $\sigma_i = \sigma$.

In addition to Gaussians (4.1), other nuclear functions can be used, for example, B-splines that meet the unit breakdown condition, paired wavelets, flexible activation-belonging functions [18, 19], etc.

In the second hidden layer, aggregated values are calculated $\prod_{i=1}^{n} \mu_{li}(x_i(k))$, while for Gaussians with the same width parameters:

$$\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)=\prod_{i=1}^{n}\exp\left(-\frac{\left(x_i(k)-c_{li}\right)^2}{2\sigma^2}\right)=\exp\left(\frac{\left\|x(k)-c_l\right\|^2}{2\sigma^2}\right),$$

where $c_l=\left(c_{l1},c_{l2},\ldots,c_{ln}\right)^{\mathrm{T}}$.

Thus, the signals at the outputs of the multiplication blocks of the second hidden layer are similar to the signals at the outputs of the neurons of the first hidden layer of conventional radial basis function networks (RBFN) [20].

The outputs of the third hidden layer are values $w_l\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)$, the fourth is $\sum_{l=1}^{h}w_l\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)$ and $\sum_{l=1}^{h}\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)$, finally, a signal appears at the output of the system (of the fifth output layer):

$$\hat{y}\left(x(k)\right)=\frac{\sum_{l=1}^{h}w_l\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}{\sum_{l=1}^{h}\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}=\sum_{l=1}^{h}w_l\frac{\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}{\sum_{l=1}^{h}\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}=\sum_{l=1}^{h}w_l\varphi_l\left(x(k)\right)=w^{hT}\varphi^h\left(x(k)\right),$$

where

$$\varphi_l\left(x(k)\right)=\frac{\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)}{\sum_{l=1}^{h}\prod_{i=1}^{n}\mu_{li}\left(x_i(k)\right)},\ w^h=\left(w_1,w_2,\ldots,w_2\right)^{\mathrm{T}},$$

$$\varphi^h\left(x(k)\right)=\left(\varphi_1\left(x(k)\right),\varphi_2\left(x(k)\right),\ldots,\varphi_h\left(x(k)\right)\right)^{\mathrm{T}}.$$

It is easy to see that the considered system implements a non-linear mapping of the input space into a scalar output signal similar to the normalized RBFN [21] and its architecture (at fixed h) coincides with the zero-order ANN system (Takagi, Sugeno, Kang'a), that is, the Wang-Mendel architecture [22].

In order to understand the interrelationship of the proposed method in the process of ANN operation, the authors built a typical ANN operation algorithm with the ANN training method (**Fig. 4.2**). The initial stage is the input of the initial data, namely the initial architecture and parameters of the artificial neural network (Steps 1–4).

Step 1. The determination of the number of layers in the artificial neural network.

Step 2. The determination of the number of nodes in the artificial neural network layer.

Step 3. Determination of the number of connections between layers and nodes of the artificial neural network.

Step 4. Determination of the number of hidden layers of the artificial neural network.

Step 5. Determination of the number and parameters of the membership function (MF).

Step 6. Determination of synaptic weights of artificial neural network connections.

Step 7. Verification of compliance of the architecture and parameters of the artificial neural network with the proposed requirements.

Step 8. Making a decision to adjust the architecture and parameters of the artificial neural network.

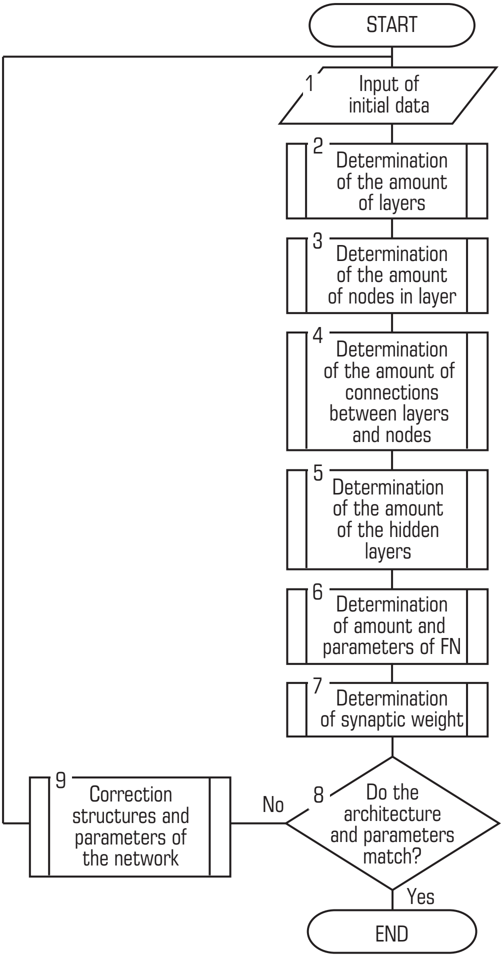

Fig. 4.2 The algorithm of the functioning and learning of an evolving artificial neural network

Let's consider in detail the stages of the proposed teaching method:

Step 5. Determination of the number and parameters of the membership function.

The process of setting parameters and the number of membership functions is as follows. Let the input of the system, which in the initial state in the first hidden layer lacks membership functions, receive the first observation of the training sample $x(1)=\left(x_1(1),x_2(1),\ldots,x_n(1)\right)^T$.

This observation forms the first node of the first hidden layer μ_1 so $c_{1i}(1)=x_i(1)$. In this way, n membership functions are created and a single synaptic weight $w_1(0)$ is formed, which is given randomly in the interval $-1\leq w_1(0)\leq 1$.

Then for this membership function μ_1 the neighborhood radius r determined by the maximum possible number of the membership function in the NFS h is given. So, if the membership functions along the axes are uniformly distributed, then:

$$r=\frac{2}{h-1}. \tag{4.2}$$

Then when the second observation $x(2)$ is received the condition is checked:

$$\max_i\left|c_{1i}-x_i(2)\right|\leq r. \tag{4.3}$$

If this condition is met, the centers of the node's μ_1 membership functions are corrected according to the rule:

$$c_{1i}(2)=c_{1i}(1)+\eta(2)\left(x_i(2)-c_{1i}(1)\right), \tag{4.4}$$

where η is the learning step parameter. Thus, at $\eta(2)=0.5$:

$$c_{1i}(2)=\frac{c_{1i}(1)+x_i(2)}{2}. \tag{4.5}$$

If condition (4.3) is not fulfilled, then the second node μ_2 is formed membership functions of the first hidden layer, the centers of which:

$$c_{2i}(2)=x_i(2). \tag{4.6}$$

Simultaneously with the node μ_2 in NFS, a second synaptic weight w_2 is introduced, which is also given randomly. Let until the moment of receipt of the input NFS observation formed membership function $\mu_1,\mu_2,\ldots,\mu_p$, $p<h$ of nodes p, with centers $c_{li}(k-1)$, $l=1,2,\ldots,p$; $i=1,2,\ldots,n$. With receipt $x(k)$, the condition is checked:

$$\max_i\left|c_{li}-x_i(k)\right|\leq r,\ \forall l=1,2,\ldots,p. \tag{4.7}$$

If this condition is met, the centers of the membership functions closest to the corresponding components $x(k)$ are corrected according to the rule:

$$c_{li}(k)=c_{li}(k-1)+\eta(k)\left(x_i(k)-c_{li}(k-1)\right). \tag{4.8}$$

It is easy to see that (4.8) is nothing but the self-learning rule of T. Kohonen "The winner takes all" [23] with the only difference that the self-learning of the Kohonen map is implemented on the hypersphere:

$$\left\|x(k)\right\|_2=1, \tag{4.9}$$

and rule (4.8) is on the hypercube:

$$\left\|x(k)\right\|_\infty=1. \tag{4.10}$$

In the event that condition (4.7) is not fulfilled, the system is formed $(p+1)$ node $(p+1\le h)$ with centers of belonging functions:

$$c_{p+1,i}(k)=x_i(k). \tag{4.11}$$

Simultaneously with the node μ_{p+1}, synaptic weight w_{p+1} is formed.

As can be seen, the mentioned procedure is a hybrid algorithm of evolving N. Kasabov [24] and self-organizing maps of T. Kohonen [23]. At the same time, the process of evolution of the architecture-self-learning of membership functions can take place both continuously and until the number of membership functions is reached.

Tuning of the center and width parameters of the membership functions can be done by a tutored algorithm based on the minimization of the objective function.

Tuning is typically specified using the Euclidean norm and for one pair of training data $\left(x(k),y(k)\right)$ which has the form:

$$E(k)=\frac{1}{2}\left(y(k)-\hat{y}(k)\right)^2=\frac{1}{2}\left(y(k)-w^T\varphi\left(x(k)\right)\right)^2. \tag{4.12}$$

While applying the fastest descent method, the corresponding adaptation formulas in the general case for $(n\times 1)$-dimensional vector of input signals takes the form:

$$c_{rj}(k+1)=c_{rj}(k)-\eta_c\frac{\partial E(k)}{\partial c_{rj}}, \tag{4.13}$$

CHAPTER 4

$$\sigma_{rj}(k+1)=\sigma_{rj}(k)-\eta_{\sigma}\frac{\partial E(k)}{\partial \sigma_{rj}}, \tag{4.14}$$

where η_c is the learning step parameter for the membership function center parameters; η_σ is the learning step parameter for the membership function width parameter; $r=1,2,...,h$, $j=1,2,...,n$.

To simplify derivative calculations and speed up the calculation of the value of the membership function, the formula for adapting the width parameter can be written in the form:

$$-0.5\sigma_{rj}^{-2}(k+1)=\sigma_{rj}(k)-\eta_{\sigma}\frac{\partial E(k)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}. \tag{4.15}$$

When traditional Gaussians (4.9) are used as membership functions, the corresponding formulas for the gradient of the objective function (4.12) for one pair of training data take the form:

$$\frac{\partial E(k)}{\partial c_{rj}}=\left(w^{\mathrm{T}}\varphi(x(k))-(x(k))\right)\sum_{l=1}^{h}w_l\frac{\partial\varphi_l(x(k))}{\varphi c_{rj}}, \tag{4.16}$$

$$\frac{\partial E(k)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}=\left(w^{\mathrm{T}}\varphi(x(k))-y(x(k))\right)\sum_{l=1}^{h}w_l\frac{\partial\varphi_l(x(k))}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}, \tag{4.17}$$

where

$$\varphi_l(x(k))=\frac{\prod_{i=1}^{n}\mu_{li}(x_i(k))}{\sum_{p=}^{h}\prod_{i=1}^{n}\mu_{pi}(x_i(k))}. \tag{4.18}$$

Derivatives $\partial\varphi_l(x)/\partial c_{rj}$ and $\partial\varphi_l(x)/\partial\left(-0.5\sigma_{rj}^{-2}\right)$, determined on the basis of (4.9) and (4.18), can be written as:

$$\frac{\partial\varphi_l(x)}{\partial c_{rj}}=\frac{\delta_{lr}m(x)-t_l(x)}{(m(x))^2}\prod_{i=1,i\neq j}^{n}\mu_{ri}(x_i)\frac{\partial\mu_{ri}(x_j)}{\partial c_{rj}}, \tag{4.19}$$

$$\frac{\partial\varphi_l(x)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}=\frac{\delta_{lr}m(x)-t_l(x)}{(m(x))^2}\prod_{i=1,i\neq j}^{n}\mu_{ri}(x_i)\frac{\partial\mu_{ri}(x_j)}{\partial\left(-0.5\sigma_{rj}^{-2}\right)}, \tag{4.20}$$

where δ_{lr} is the Kronecker's delta; $t_l(x)=\prod_{i=1}^{n}\mu_{li}(x_i)$, $m(x)=\sum_{p=1}^{h}\prod_{i=1}^{n}\mu_{pi}(x_i)$.

Derivatives $\partial\mu_{rj}(x_j)/\partial c_{rj}$ and $\partial\mu_{rj}(x_j)/\partial\left(-0.5\sigma_{rj}^{-2}\right)$ are determined on the basis of (4.9), can be written as:

$$\frac{\partial\mu_{rj}(x_j)}{\partial c_{rj}}=\frac{x_j-c_{rj}}{\sigma_{rj}^2}\exp\left(-\frac{(x_j-c_{rj})^2}{2\sigma_{rj}^2}\right), \tag{4.21}$$

$$\frac{\partial\mu_{ri}(x_j)}{\partial(-0.5\sigma_{rj}^{-2})}=(x_j-c_{rj})^2\exp\left(-\frac{(x_j-c_{rj})^2}{2\sigma_{rj}^2}\right). \tag{4.22}$$

Step 6. Determination of synaptic weights of artificial neural network connections.

As already noted, well-known learning-identification algorithms can be used to adjust the synaptic weights of the neuro-fuzzy system [26, 27]:

– exponentially weighted recurrent least squares method:

$$\begin{cases} w^h(k)=w^h(k-1)+\dfrac{P^h(k-1)\left(y(k)-w^{hT}(k-1)\varphi^h(x(k))\right)}{\beta+\varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\varphi^h(x(k))= \\ =w^h(k-1)+\dfrac{P^h(k-1)\left(y(k)-\hat{y}^h(x(k))\right)}{\beta+\varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\varphi^h(x(k)); \\ P^h(k)=\dfrac{1}{\beta}\left(P^h(k-1)-\dfrac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta+\varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))}\right)= \\ =\left(\sum_{r=1}^{k}\varphi^h(x(\tau))\varphi^{hT}x(\tau)\right)^{-1},\ 0<\beta\le 1, \end{cases} \tag{4.23}$$

where $y(t)$ is the external training signal; β is the option to forget outdated information;

– speed-optimized gradient one-step Kachmaj-Widrow-Hoff algorithm:

$$w^h(k)=w^h(k-1)+\frac{y(k)-w^{hT}(k-1)\varphi^h(x(k))}{\left\|\varphi^h(x(k))\right\|^2}\varphi^h(x(k)); \tag{4.24}$$

– a learning algorithm with both tracking and smoothing properties [27]:

$$\begin{cases} w^h(k)=w^h(k-1)+\left(\beta^h(k)\right)^{-1}\left(y(k)-w^{hT}(k-1)\varphi^h(x(k))\right)\varphi^h(x(k)); \\ \beta^h(k)=\beta\beta^h(k-1)+\left\|\varphi^h(x(k))\right\|^2,\ 0\le\beta\le 1, \end{cases} \tag{4.25}$$

and similar procedures.

Procedure (4.25) is related to algorithm (4.23) by the relation:

$$\beta^h(k) = TrP^h(k), \tag{4.26}$$

and at $\beta = 0$ let's obtain the form of algorithm (4.23).

The process of adjusting the synaptic weights can occur simultaneously with the self-learning-evolution of the first hidden layer.

Let until the moment of observation $x(k)$ of p nodes of membership functions $\mu_1, \mu_2, \ldots, \mu_p$ are formed and calculated vector of synaptic weights $w^p(k-1)$. Let the condition (4.7) not be fulfilled, which immediately leads to the formation of a node μ_{p+1} and assigning an arbitrary initial value of the synaptic weight w_{p+1}. At the same time, the NFS output signal can be presented in the form:

$$\hat{y}^{p+1}(x(k)) = \left(w^{p+1}(k-1)\right)^{\mathrm{T}} \varphi^{p+1}(x(k)) = w^{p\mathrm{T}}(k-1)\varphi^p(x(k)) + w_{p+1}\varphi_{p+1}(x(k)), \tag{4.27}$$

and algorithm (4.24) is:

$$\begin{cases} w^{p+1}(k) = \left(\dfrac{w^p(k-1)}{w_{p+1}}\right) + \left(\beta^{p+1}(k)\right)^{-1}\left(y(k) - \hat{y}^{p+1}(x(k))\right)\left(\dfrac{\varphi(x(k))}{\varphi_{p+1}(x(k))}\right); \\ \beta^{p+1}(k) = \beta\beta^p(k-1) + \left\|\varphi^p(x(k))\right\|^2 + \varphi_{p+1}^2(x(k)). \end{cases} \tag{4.28}$$

As it is possible to see, the process of simultaneous evolution-self-learning-supervised learning does not cause any computational problems.

Step 7. Verification of compliance of the architecture and parameters of the artificial neural network with the proposed requirements.

At this step, the compliance of the architecture and parameters of the artificial neural network with the requirements is checked. The specified requirements for the functioning of an artificial neural network are put forward at the stage of designing decision making support systems and depend on the type and volume of tasks performed by decision making support systems.

Step 8. Making a decision to adjust the architecture and parameters of the artificial neural network.

On the basis of a comparative assessment of the requirements for the operation of ANN and their real effectiveness, a decision is made to adjust the parameters of ANN, namely (Steps 6–8 in the scheme of **Fig. 4.2**):

– the type and parameters of the membership function;

– synaptic weights between connections.

If it is impossible to meet the necessary requirements for the ANN, a decision is made to change the architecture of the ANN and determine the initial parameters of the ANN (Steps 1–5 in the scheme of **Fig. 4.2**).

A method of learning artificial neural networks for intelligent decision making support systems is proposed. The operation of the proposed method was simulated in the MathCad 14 software environment.

The effectiveness of the proposed multi-layer neuro-phase system evolving with hybrid learning was demonstrated while evaluating a radio communication system with pseudo-random tuning of the operating frequency (PRTOF) under the influence of intentional noise interference.

The research of the developed method showed that the specified training method provides an average of 16–23 % higher efficiency of training artificial neural networks and does not accumulate errors during training.

4.2 THE DEVELOPMENT OF KOHONEN METHOD OF LEARNING ARTIFICIAL NEURAL NETWORKS WITH AN EVOLVING ARCHITECTURE

The Kohonen network [2] refers to self-organizing networks. This means that they do not get the desired output signal when the input training vector arrives, and as a result of training, the network divides the input signals into classes, thus forming topological maps.

One of the most important properties of a trained Kohonen network is its ability to generalize. The essence of the Kohonen network is that the input vectors are clustered into groups of similar vectors. At the same time, the weights of the network are adjusted so that the input images belonging to the same cluster activate the same output neuron.

The vector of each of the neurons of the Kohonen network can replace a group of corresponding classified vectors. Thus, this property allows to use this type of network in the field of data compression.

It is worth noting that T. Kohonen self-organizing map implements the mapping of the input space of dimension n into the output space of dimension m.

The self-organized map has a very simple architecture with direct information transfer. In addition to the zero (receptor) layer, it contains a single layer of neurons, which is very often called Kohonen layer [2].

Thanks to such an organization, each neuron of the network receives all the information about the analyzed image and generates a corresponding response at its output. After that, a competition mode occurs in the Kohonen layer, as a result of which a single winning neuron with the maximum output signal is determined. This signal through lateral connections ensures the excitation of the nearest "neighbors" of the winner and the suppression of the reaction of distant nodes. Self-organizing maps can have different topologies. However, receptors and neurons are most often located in the nodes of a one- or two-dimensional lattice.

Let's consider the architecture of a self-organizing map in more detail. An n-dimensional input signal is received at the input of the network. The network contains a single layer of m neurons that form rectangular lattices on the plane.

Neurons are characterized by their location in the network. Each neuron of the Kohonen layer is connected to each input of the zero (input) layer by direct connections and to all other neurons by transverse connections.

Fig. 4.3 presents Kohonen 1D map.

In the learning process, neighboring neurons influence each other more strongly than those located further away. It is the lateral connections in the network that ensure the excitation of some neurons and the inhibition of others.

Each neuron from the Kohonen layer forms a weighted sum of signals $f(x,w)=\sum_{i=1} w_i x_i$. At the same time, if the synapses accelerate, then $w_{ij}>0$. If the synapses are inhibitory, then $w_{ij}<0$.

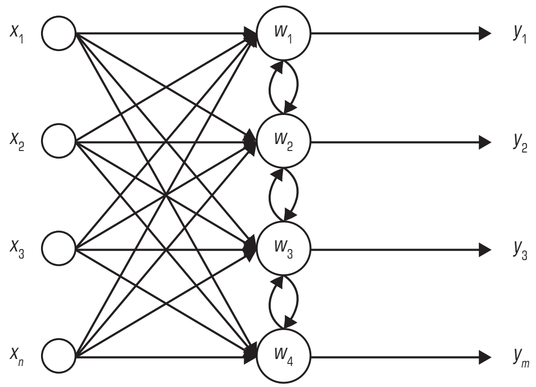

Fig. 4.3 Kohonen 1D map

Fig. 4.4 shows the proposed artificial neural network learning algorithm. The improvement of the specified learning algorithm consists in adding Steps 5–8 to the known methods of learning artificial neural networks.

Thus, additional training of artificial neural networks takes place, which was not taken into account in the works [1–17]:

– an architecture of artificial neural networks depending on the amount of input information (the number of layers, the number of hidden layers, the number of connections between neurons in a layer and between layers);

– the quantities and parameters of the membership function;

– it does not require storage of previously calculated data.

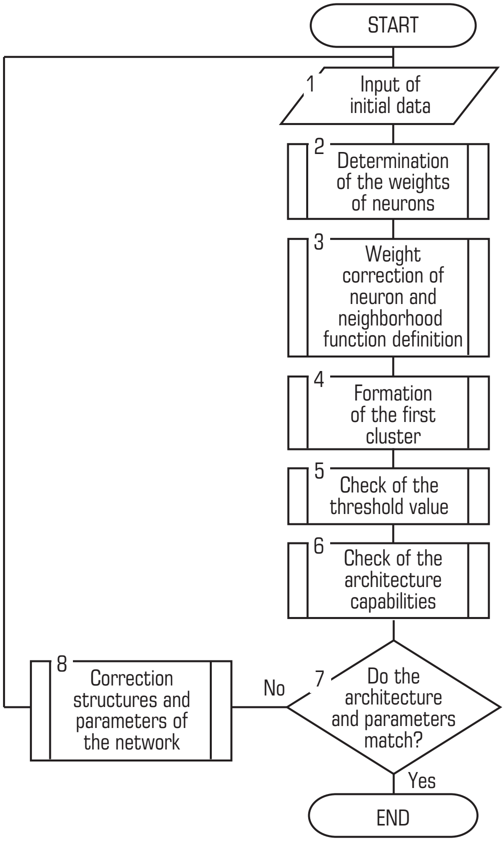

Fig. 4.4 An algorithm of the functioning and learning of an evolving artificial neural network

Step 1. The initial stage is the initialization of the initial values of the synaptic weights.

Step 2. Determination of neuron weights.

Step 3. Correction of neuron weights and definition of neighborhood function.

Step 4. Formation of the first cluster.

Step 5. Checking the threshold value.

Step 6. Checking the capabilities of the architecture to process the amount of information coming to its input.

Step 7. System architecture evolution.

Before the Kohonen network learning algorithm starts, the input vectors are pre-normalized:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|},\ i = 1,2\ldots,N. \tag{4.29}$$

The Kohonen network learning algorithm itself can be described as a sequence of steps:

Step 1: the initial values of the synaptic weights $w_{ij} = 0$ are initialized.

One of the frequently used methods of initialization is the assignment of synaptic weights to values equal to randomly selected vectors from a set of observations. In general, there are three main ways of initializing initial weights:

– initialization with random values, when all weights are given small random values;

– initialization with examples, when values of randomly selected examples from the training sample are set as initial values;

– linear initialization. In this case, the weights are initialized by the values of the vectors linearly ordered along the linear subspace passing between the two main vectors of the original data set.

Step 2: a normalized vector of signals is applied to the input of the system and the weight vector (neuron) closest to $\tilde{x}$ is selected that is, the vector for which the Euclidean distance to x will be the smallest:

$$\arg\ \min_j \|\tilde{x} - w_j\|,\ \ j=1,2,\ldots,l. \tag{4.30}$$

This expression can also be written in the following form:

$$c = \arg\ \max\left(\tilde{x}^{\mathrm{T}} w_j\right),\ \ j=1,2,\ldots,l. \tag{4.31}$$

Step 3: correction (adjustment) of the vector of synaptic weights w_{ij} is performed as a rule:

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k) f_{ij}(k)\left(\tilde{x}(k) - w_{ij}(k)\right), \tag{4.32}$$

where $w_{ij}(k+1)$ is the new weight value; $\eta(k)$ is the amplification factor changes over time (on the first iteration $\eta = 1$ gradually decreases to zero, that is, $\eta(k) \in (0,1]$); $f_{ij}(k)$ is a monotonically decreasing function (neighborhood function) of the species:

$$f_{ij}(k) = f\left(\|r_i - r_j\|, k\right), \tag{4.33}$$

where r_i is a vector that determines the position of the i-th neuron in the grid; r_j is a vector that determines the position of the j-th neuron in the lattice.

It is obvious that for the winning neuron $f_c\left(\|r_i - r_j\|\right) = f_c(0) = 1$.

Most often, the Gaussian, "Mexican hat", cosine function, Epanechnikov function, etc. are chosen as the neighborhood option. At the same time, the output signals of the network are defined as follows:

$$y_j(k)=\begin{cases}1, \text{ if } w_j^{\mathrm{T}}(k)\tilde{x}(k)>w_p^{\mathrm{T}}(k)\tilde{x}(k) \text{ for all } p\neq j;\\ 0, \text{ otherwise.}\end{cases} \tag{4.34}$$

Then, *Steps 1* and *2* are repeated until the output values of the network stabilize with the specified accuracy. The meaning of adjustment according to expression (4.33) is reduced to the fact that the difference between the input vector is minimized and the vector of synaptic weights of the winning neuron.

In other words, it is possible to say that this algorithm "pulls" the vector of synaptic weights of the winning neuron to the current input image during the tuning process $\tilde{x}(k)$. In this case, there is a vector $\tilde{x}(k)-w_{ij}(k)$, which then decreases by an amount $\eta(k)$, which sets the learning rate. Thus, it is possible to say that learning is reduced to the rotation of the neuron's weight vector in the direction of the input vector without significantly changing its length.

In the work [18], an original online evolving fuzzy clustering method (EFCM) is proposed, which is based on a probabilistic approach to solving the problem. At the same time, the main parameter that ultimately determines the final result is the radius of the formed clusters, chosen based on empirical considerations and ultimately determines the number of possible classes. Despite the effectiveness of probabilistic fuzzy clustering method (FCM), their "weak" point is a "hard" limitation:

$$\sum_{j=1}^{m}u_j(k)=1,\ \forall k=1,2,\dots,N. \tag{4.35}$$

The so-called possibilistic fuzzy method (PCM) to fuzzy clustering [17], associated with the optimization of the objective function, is devoid of this drawback. At the same time, the observation also belongs to all classes, that is, equidistant from all centroids, but does not belong to any of the clusters:

$$E(u_j,c_j)=\sum_{k=1}^{N}\sum_{j=1}^{m}u_j^{\beta}(k)\left\|x(k)-c_j\right\|^2+\sum_{j=1}^{m}\mu_j\sum_{k=1}^{N}\left(1-u_j(k)\right)^{\beta}, \tag{4.36}$$

where $c_j=\left(c_{j1},c_{j2},\dots,c_{jn}\right)^{\mathrm{T}}$ is the center of gravity of the j-th cluster, which is calculated during data processing; $\beta>1$ is the fuzzification parameter ("fuzzifier"), which determines the "blurring" of the boundaries between clusters and usually rely on $\beta=2$; $\mu_j>1$ is a scalar parameter that determines the distance at which the membership takes the value 0.5, i.e. if:

$$\left\|x(k)-c_j\right\|^2=\mu_j, \tag{4.37}$$

then $\mu_j(k)=0.5$.

CHAPTER 4

Using a probabilistic approach leads to an evolutionary fuzzy clustering method (EPCM), which is conveniently written in the form of a sequence of steps:

Step 4: when an observation $x(1)$ is received the first cluster with a centroid is formed c_1.

Step 5: upon receipt of observation the condition is checked:

$$\left\|x(2)-c_1\right\| \leq \Delta, \tag{4.38}$$

where Δ is some threshold priori set.

If this condition is met, then observation $x(1)$ does not form a new center of gravity, so it was required that it belongs to the first cluster with the membership level:

$$u_1(2)=\left(1+\left(\frac{\left\|x(2)-c_1\right\|^2}{\mu_1}\right)^{\frac{1}{\beta-1}}\right)^{-1}. \tag{4.39}$$

If the condition is fulfilled:

$$\Delta<\left\|x(2)-c_1\right\| \leq 2\Delta, \tag{4.40}$$

then the centroid c_1 is corrected according to Kohonen WTA (winner-takes-all, WTA) self-learning rule [18]:

$$c_1(2)=c_1(1)+\eta(2)\left(x(2)-c_1(1)\right), \tag{4.41}$$

where $\eta(2)$ is the tuning step parameter.

At the same time, the centroid c_1 is pulled to the vector of observations $x(1)$; if for the $x(1)$ inequality holds:

$$2\Delta<\left\|x(2)-c_1\right\|, \tag{4.42}$$

then a second cluster with a centroid is formed:

$$c_2=x(2). \tag{4.43}$$

At the same time, the appropriateness levels should be calculated and according to the formulas below.

Step 6: checking the capabilities of the architecture to process the amount of information coming to its input.

So, if there are N observations and m clusters with centroids, calculations of all properties and adjusted coordinates of centroids are evaluated according to the ratio:

$$\begin{cases} u_j(k) = \left(1 + \left(\dfrac{\left\| x(k) - c \right\|^2}{\mu_j}\right)^{\frac{1}{\beta - 1}}\right)^{-1}; \\ c_j = \dfrac{\sum_{k=1}^{N} u_j^{\beta}(k) x(k)}{\sum_{k=1}^{N} u_j^{\beta}(k)}; \\ \mu_j(k) = \dfrac{\sum_{k=1}^{N} u_j^{\beta}(k) \left\| x(k) - c_j \right\|^2}{\sum_{k=1}^{N} u_j^{\beta}(k)}, \end{cases} \quad (4.44)$$

let's obtain as a result of minimization (4.36) for all evaluated parameters.

The system of equations (4.44) is essentially a batch algorithm for processing information so that when an observation $x(N+1)$ is received, all calculations must be performed anew. It is clear that with a sufficiently high frequency of data arrival, the approach may turn out to be ineffective.

For this purpose, it is necessary to develop recurrent procedures that do not require the storage of previously processed data. The mentioned recurrent procedures can be implemented on the basis of a two-layer adaptive neuro-fuzzy network with the following architecture.

The first hidden layer of the network is formed by ordinary Kohonen neurons N_j^K, connected by lateral ties, through which the process of competition is implemented. The output layer of the network, formed by nodes N_j^u, designed to calculate the membership levels of each observation $x(k)$ to each j-th cluster, $j = 1,2,3,\ldots,m$. To adjust the cluster centroids, a recurrent self-learning procedure is used, which has the form [10]:

$$\begin{cases} c_j(k+1) = c_j(k) + \dfrac{u_j^{\beta}(k)}{k+1}\left(x(k+1) - c_j(k)\right); \\ u_j(k+1) = \dfrac{1}{1 + \left(\dfrac{\left\| x(k+1) - c_j(k+1) \right\|^2}{\mu_j(k)}\right)^{\frac{1}{1-\beta}}}; \\ \mu_j(k+1) = \dfrac{\sum_{p=1}^{k+1} u_j^{\beta}(p) \left\| x(p) - c_j(k+1) \right\|^2}{\sum_{p=1}^{k+1} u_j^{\beta}(p)}. \end{cases} \quad (4.45)$$

It is easy to see that the first expression (4.45) is a self-learning WTM rule with a narrowing neighborhood function $(k+1)^{-1}u_j^{\beta}(k)$.

Step 7: an evolution of the system architecture.

The system evolution process, similar to the previous one, begins with a single Kohonen neuron, which specifies the coordinates of the first centroid c_1. The next neuron is added to the network when condition (4.42) is fulfilled, which in this case takes the form:

$$2\Delta < \left\| x(k) - c_1(k-1) \right\|. \tag{4.46}$$

At this moment, a neuron with a centroid $c_2(k) = x(k)$ is formed. It should be noted here that since in Kohonen neural networks the data is pre-normalized to the hyperlayer so that:

$$\left\| x(k) \right\|^2 = \left\| c_j(k) \right\|^2 = 1, \tag{4.47}$$

inequality (4.14), which determines the need to introduce new neurons to the network, takes shape:

$$-1 \le 1 - 2\Delta^2 < c_j^T(k-1)x(k) \le 1, \ \forall j = 1,2,\ldots,m, \tag{4.48}$$

or

$$-1 \le 1 - 2\Delta^2 < \cos\left(c_j(k-1)x(k)\right) \le 1. \tag{4.49}$$

Thus, the building up of the architecture occurs as a result of constant control of inequalities (4.48) or (4.49), which occurs in the event that these inequalities are violated.

Thanks to the use of the possible approach, it is expedient to implement another "branch of evolution", namely, if at some point in time it turns out that the appropriateness of the observation $x(k)$ do not exceed some threshold value:

$$u_j(k) < \varepsilon, \ \forall j = 1,2,\ldots,m, \tag{4.50}$$

thus, the observation $x(k)$ located far enough from all already formed centroids, this can also serve as a signal for the formation of a new cluster:

$$c_{m+1}(k) = x(k). \tag{4.51}$$

To assess the quality of fuzzy clustering, the popular Xi-Beni index [10] in an extended form [14] can be used.

For a fixed sample with N observations, this index has the form:

$$EXB(N)=\frac{\left(\sum_{k=1}^{N}\sum_{j=1}^{m}u_j^{\beta}(k)\left\|x(k)-c_j(N)\right\|^2 \Big/ N\right)}{\min_{j\neq l}\left\|c_j(N)-c_l(N)\right\|^2}=\frac{NEXB(N)}{DEXB(N)}. \quad (4.52)$$

For online processing, this index, similar to the centroids of clusters, can also be calculated recursively:

$$EXB(k+1)=\frac{NEXB(k+1)}{DEXB(k+1)}=$$
$$=\frac{NEXB(k)+\frac{1}{k+1}\left(\sum_{j=1}^{m}u_j^{\beta}(k+1)\left\|x(k+1)-c_j(k+1)\right\|^2-NEXB(k)\right)}{\min_{j\neq l}\left\|c_j(k+1)-c_l(k+1)\right\|^2}. \quad (4.53)$$

Adding expression (4.52) to the training procedure (4.47) will allow to organize additional control over the number of clusters formed by the system. Thus, the introduction of the third threshold and checking the condition at each step:

$$EXB(k+1)>\delta, \quad (4.54)$$

it is possible to stop the process of building up neurons in case of violation of inequality (4.54).

A two-dimensional artificially generated data set was used for the experiment. It contained 15 clusters with different levels of overlap. The data sample contained 5000 observations. Data were submitted for processing in sequential mode. To compare the quality of clustering, FCM and a system based on the Evolving Fuzzy Clustering Method (EFCM) with different values of the threshold parameter were used.

The Xi-Beni index was used as a criterion for assessing the quality of clustering.

Table 4.1 presents the comparative results of clustering.

Table 4.1 Comparative clustering results

The system	The number of clusters	Algorithm parameters	XB (Xi-Beni index)	Clustering time, s
FCM (Fuzzy C-Means)	15	–	0.0903	2.69
EFCM	9	Dthr=0.24	0.1136	0.14
EFCM	12	Dthr=0.19	0.1548	0.19
The proposed system (batch mode)	12	delta=0.1	0.0978	0.37
The proposed system (online mode)	12	delta=0.1	0.1127	0.25

To conduct the next experiment, a sample of data describing the characteristics of the monitoring object was used. The data sample contained 210 observations (**Table 4.2**). Before clustering, the features of the observations were normalized to the interval $[0,1]$.

Table 4.2 Comparative clustering results

The system	The number of clusters	Algorithm parameters	XB (Xi-Beni index)	Time, s
FCM (Fuzzy C-Means)	3	–	0.1963	0.81
EFCM	4	Dthr=0.6	0.2330	0.04
The proposed system (batch mode)	3	delta=0.4	0.2078	0.45
The proposed system (online mode)	3	delta=0.4	0.2200	0.30

The parameters of the analyzed systems and the number of detected clusters, are also given in the **Table 4.2**. The Xi-Beni (XB) index was used to assess the quality of the systems. It is worth noting that the proposed system showed a better PC (partition coefficient) result in comparison with EFCM and a better result in terms of operation time compared to FCM. Both the proposed systems and FCM identified three fuzzy clusters.

The research of the developed method showed that the specified training method provides an average of 10–18 % higher efficiency of training artificial neural networks and does not accumulate errors during training (**Tables 4.1, 4.2**).

4.3 THE DEVELOPMENT OF A LEARNING METHOD FOR CASCADED ARTIFICIAL NEURAL NETWORKS WITH AN EVOLVING ARCHITECTURE

The architecture of the evolving cascade ANN [16–18] is presented in **Fig. 4.5**.

The zero layer of the system receives the $(n\times 1)$-dimensional vector of input signals $x(k)=$ $=(x(k), x_1(k), x_2(k), \ldots, x_n(k))^{\mathrm{T}}$, which is then transmitted to the first hidden layer c_n^2, containing nodes-neurons, each of which has two inputs.

Output signals are formed at the outputs of nodes $N^{[1]}$ of the first hidden layer $\hat{y}_s^{[1]}$ $s=1,2,\ldots,0.5n(n-1)=c_n^2$. Then, these signals are sent to the *SB* selection block, which performs the function of sorting the nodes of the first hidden layer according to the accepted criterion (most often according to the value of the mean square of the error $\sigma^2_{\hat{y}_s^{[1]}}$), so $\sigma^2_{\hat{y}_1^{[1]*}}<\sigma^2_{\hat{y}_2^{[1]*}}<\ldots<\sigma^2_{\hat{y}_s^{[1]*}}$. Selection block outputs $\hat{y}_1^{[1]*}$ and $\hat{y}_2^{[1]*}$ enter the input of a single node-neuron of the second layer, at the output of which the output signal $\hat{y}^{[2]*}$ is formed. This output signal together with the output signal of the selection block $\hat{y}_3^{[1]*}$ enters the input of the node-neuron of the next layer. The process of increasing the cascades continues until the necessary accuracy of information processing is achieved.

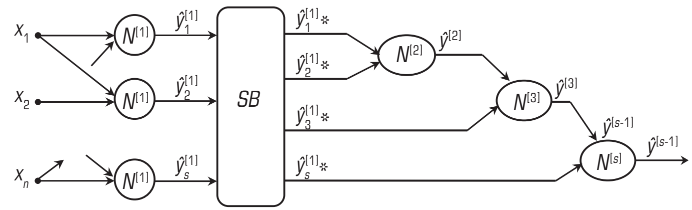

Fig. 4.5 The architecture of the evolving cascade ANN

Two-input neuro-fuzzy systems, discussed earlier [1] and two-input neuro-fuzzy nodes, the architecture of which will be discussed further, can be used as nodes of the considered evolving cascade ANN.

The problem of ANN training is clearly complicated if the data entering the neural network input is non-stationary and non-linear, contains quasi-periodic components, stochastic and chaotic components. In these conditions, nonlinear models based on the mathematical apparatus of computational intelligence [2–4] and, first of all, neuro-fuzzy systems have shown themselves to be the best. The advantages are due to high approximating and extrapolating properties, ability to learn, transparency and interpretability of the obtained results. The so-called NARX-models, which have the form, should be highlighted here:

$$\hat{y}(k)=f\left(y(k-1),y(k-2),\dots,y(k-n_y),x(k-1),\dots,x(k-n_x)\right), \tag{4.55}$$

where $\hat{y}(k)$ is the estimation (forecast) of the sequence at a discrete time point $k=1,2,\dots$; $f(\bullet)$ is some non-linear transformation implemented by a neuro-fuzzy system; $x(k)$ is an exogenous, observable factor that determines behavior $y(k)$. It can be seen that the ANARX model (Additive Nonlinear Autoregressive Exogenous – NARX) and the WANARX model (Additive Nonlinear Autoregressive Exogenous Weighted – ANARX) correspond to description (4.55). Such models are well studied, there are quite a large number of architectures and algorithms for their learning, however, it is assumed that the order of the model n_y, n_x, somehow predetermined. In the case of structural non-stationarity of the studied series, these orders are unknown a priori, and must also be adjusted during the learning process. The situation is significantly complicated if information for processing arrives with a fairly high frequency in the form of a data stream [6–9]. In this case, the most popular evolving systems are too cumbersome to learn and process information online. Considering the above, the classic procedure for learning networks is the adjustment of synaptic weights, without taking into account other possibilities of learning the network, such as the type of architecture of individual network nodes and the composition of the network (combinations of nodes).

Fig. 4.6 shows the proposed artificial neural network learning algorithm. The improvement of the specified learning algorithm consists in adding *Steps 2*, *3* and *8* to the known methods of learning artificial neural networks.

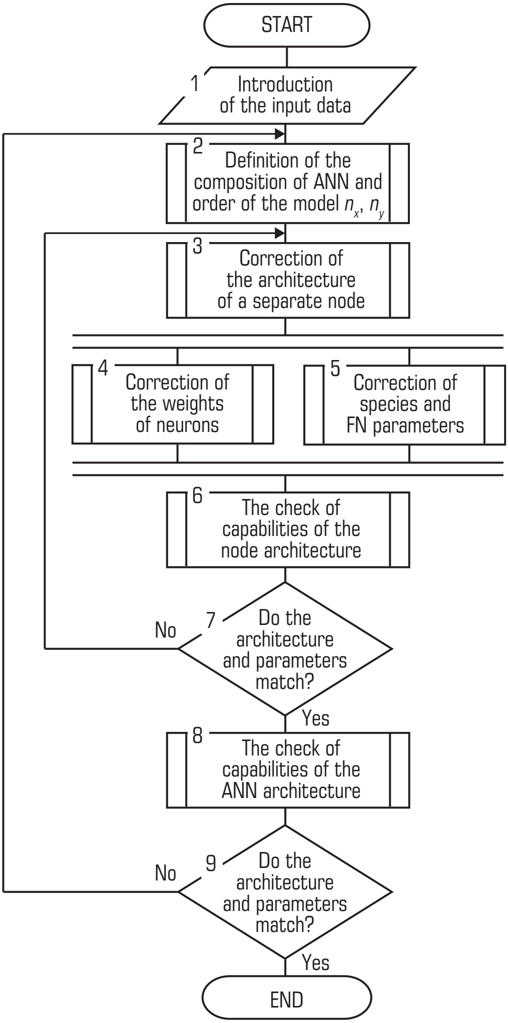

Fig. 4.6 Algorithm of functioning and training of the evolving ANN

Thus, there is additional training of artificial neural networks, which was not taken into account in the works [1–17]:

– an architecture of artificial neural networks depending on the amount of input information (the number of layers, the number of hidden layers, the number of connections between neurons in a layer and between layers);

– an architecture and parameters of a separate node of artificial neural networks;

– the possibilities of combining nodes of an artificial neural network.

Step 1. The initial stage is the entry of initial data.

Step 2. Determination of the composition of the ANN (the number of nodes) and the order of the model n_x, n_y.

Step 3. Adjusting the architecture of a separate ANN node.

Step 4. Correction of neuron weights of a separate node.

Step 5. Adjustment of the type and parameters of the membership function (FN).

Steps 6, 7. Checking the capabilities of the architecture of a separate node.

Steps 8, 9. Checking the capabilities of the architecture to process the amount of information coming to its input.

Let's consider the proposed method in detail:

Step 1. An input of initial data.

At this stage, the initial parameters of the network are entered: the number of layers, the number of nodes, the number of connections between them and the initial values of the membership function.

Step 2. Determination of the composition of the ANN (number of nodes) and order of the model n_x, n_y.

ANARX model having the form [7, 17]:

$$\hat{y}(k)=f_1\big(y(k-1),x(k-1)\big)+f_2\big(y(k-2),x(k-2)\big)+\ldots+f_n\big(y(k-n),x(k-n)\big)=$$

$$=\sum_{l=1}^{n} f_l\big(y(k-l),x(k-l)\big), \tag{4.56}$$

here $n=\max\{n_y,n_x\}$, while the original task of synthesis of the forecasting system is decomposed into a set of local tasks of parametric identification of node models with two inputs $y(k-l),x(k-l)$, $l=1,2,\ldots,n,\ldots$.

Fig. 4.7 shows the architecture of the ANARX system, formed by two lines of pure delay elements $z^{-1}\big(z^{-1}y(k)=y(k-1)\big)$ and n nodes connected in parallel $N^{[l]}$. The training of these nodes is carried out independently of each other and the introduction of new nodes or the exclusion of redundant ones does not affect the work of all other neurons, thus, the evolution of such a system is realized by elementary manipulation of the number of nodes.

Two-input neuro-fuzzy systems, considered earlier and two-input neo-fuzzy nodes are used as a node of the ANARX system under consideration.

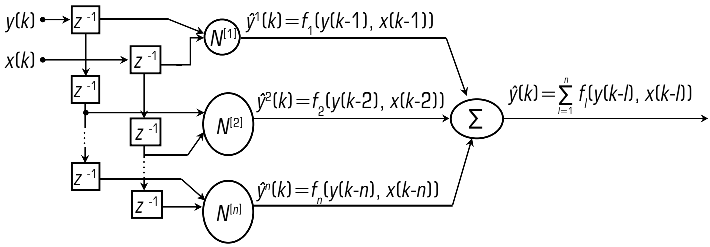

Fig. 4.7 An architecture of the ANARX system

Step 3. Adjusting the architecture of a separate ANN node.

When the speed of data processing and ease of numerical implementation of the computer intelligence system come to the fore, instead of neuro-fuzzy nodes of the ANARX model, it is advisable to use neuro-fuzzy neurons, which are nonlinear systems that learn. The architecture of the neuro-fuzzy neuron as a node of the ANARX system is shown in **Fig. 4.8**. The advantages of the neuro-fuzzy neuron include high learning speed, computational simplicity, good approximating properties and the possibility of finding the global minimum of the learning criterion in online mode.

The constituent elements of a neuro-fuzzy neuron are non-linear synapses NS_y, NS_x, in which the zero-order Takagi-Sugeno fuzzy inference rules are implemented, however, as it is easily possible to see, the neuro-fuzzy neuron is structurally much simpler than the neuro-fuzzy node shown in **Fig. 4.8**.

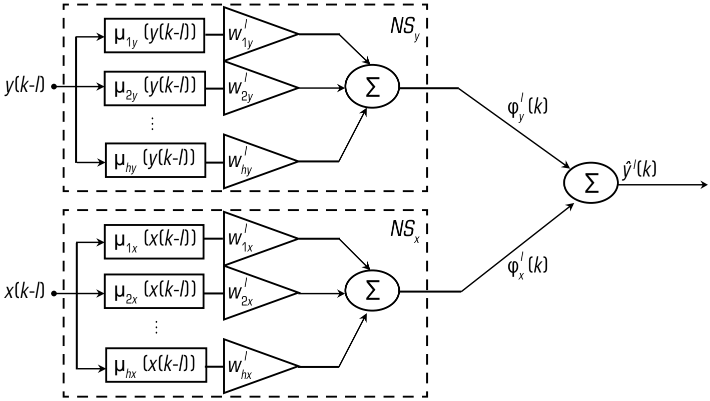

Fig. 4.8 Neo-fuzzy node of the ANARX system

While applying signals to the input of such a node $y(k-l)$, $x(k-l)$, the output value is formed at its output:

$$\hat{y}(k)=\varphi_y^l(k)+\varphi_x^l(k)=\sum_{i=1}^{h} w_{iy}^l \mu_{iy}\left(y(k-l)\right)+\sum_{i=1}^{h} w_{ix}^l \mu_{ix}\left(x(k-l)\right), \quad (4.57)$$

w_{ix}^l is the synaptic weights of the l-function neuron; μ_{ix} is the membership function.

And at the output of ANARX models as a whole:

$$\hat{y}(k)=\sum_{l=1}^{n}\left(\sum_{i=1}^{h} w_{iy}^l \mu_{iy}\left(y(k-l)\right)+\sum_{i=1}^{h} w_{ix}^l \mu_{ix}\left(x(k-l)\right)\right), \quad (4.58)$$

thus, since the neuro-fuzzy neuron is also an additive model [81], the ANARX model on neuro-fuzzy neurons is doubly additive.

Step 4. Correction of neuron weights of a separate node.

The correction of neuron weights in the ANN node is based on well-known scientific approaches, which are described, for example, in the work [18].

Step 5. Adjustment of the type and parameters of the membership function (FN).

As membership functions in neuro-fuzzy-neuron, as a rule, triangular constructions that meet the conditions of unit partitioning are used:

$$\sum_{i=1}^{h} \mu_{iy}\left(y(k-l)\right)=1; \ \sum_{i=1}^{h} \mu_{ix}\left(x(k-l)\right)=1, \quad (4.59)$$

which allows to simplify the design of the node by excluding the normalization layer from it.

In the work [1], it was proposed to use B-splines as membership functions of neuro-fuzzy-neuron, which provide a higher quality of approximation, which also meet the conditions of unit breakdown. At the same time, for a B-spline of the q-th order, it is possible to write:

$$\mu_{iy}^q\left(y(k-l)\right)=1, \begin{cases} \left.\begin{matrix} 1, \text{ if } c_{iy} \le y(k-l) < c_{i+1,y}; \\ 0, \text{ otherwise}; \end{matrix}\right\} \text{for } q=1; \\ \dfrac{y(k-l)-c_{iy}}{c_{i+q-1,y}-c_{iy}}\mu_{iy}^{q-1}\left(y(k-l)\right)+\dfrac{c_{i+q,y}-y(k-l)}{c_{i+q,y}-c_{i+1,y}}\mu_{i+1,y}^{q-1}\left(y(k-l)\right), \text{ for } q>1; \\ i=1,\ldots,h-q, \end{cases} \quad (4.60)$$

where

$$c_l=\left(c_{l1},c_{l2},\ldots,c_{ln}\right)^{\mathrm{T}};$$

$$\mu_{ix}^{q}\left(x(k-l)\right)=1,\begin{cases}\left.\begin{array}{l}1,\text{ if } c_{ix}\le x(k-l)<c_{i+1,x};\\ 0,\text{ otherwise;}\end{array}\right\}\text{ for } q=1;\\ \dfrac{x(k-l)-c_{ix}}{c_{i+q-1,x}-c_{ix}}\mu_{ix}^{q-1}\left(x(k-l)\right)+\dfrac{c_{i+q,x}-x(k-l)}{c_{i+q,x}-c_{i+1,x}}\mu_{i+1,x}^{q-1}\left(x(k-l)\right),\text{ for } q>1;\\ i=1,\ldots,h-q.\end{cases}\tag{4.61}$$

It should be noted that B-splines are a kind of generalized membership functions: for example, at $q=2$ let's get traditional triangular membership functions, at $q=4$ – cubic splines, etc.

Entering further vector variables:

$$w^{l}=\left(w_{1,y}^{l},\ldots,w_{h,y}^{l},w_{1,x}^{l},\ldots,w_{h,x}^{l}\right)^{\mathrm{T}},$$

$$\varphi^{l}(k)=\left(\mu_{1y}y(k-l)\right),\ldots,\mu_{hy}(y)(k-l),$$

$$\mu_{1x}\left(x(k-l)\right),\ldots,\mu_{hx}\left(x(k-l)\right)^{\mathrm{T}},$$

it is possible to rewrite (4.58) in the form:

$$\hat{y}^{l}(k)=w^{lT}\varphi^{l}(k),\tag{4.62}$$

and using the Kachmaj-Widrow-Hoff gradient one-step algorithm that is optimal in terms of speed for training:

$$w^{h}(k)=w^{h}(k-1)+\frac{y(k)-w^{h\mathrm{T}}(k-1)\varphi^{h}\left(x(k)\right)}{\left\|\varphi^{h}\left(x(k)\right)\right\|^{2}}\varphi^{h}\left(x(k)\right),\tag{4.63}$$

let's obtain taking into account (4.63):

$$\begin{aligned}&w^{l}(k)=w^{l}(k-1)+r_{l}^{-1}(k)\left(y(k)-w^{l\mathrm{T}}(k-1)\varphi^{l}(k)\right)\varphi^{l}(k),\\ &r_{l}(k)=\alpha r_{l}(k-1)+\varphi^{l\mathrm{T}}(k)\varphi^{l}(k),\ 0\le\alpha\le 1,\end{aligned}\tag{4.64}$$

which has both filtering and tracking properties. It can also be noticed that at $\alpha=1$ (4.63) completely coincides with the optimal Kachmaj-Widrow-Hoff algorithm (4.64).

Steps 6, 7. Checking the capabilities of the architecture of a separate node.

At this stage, the ability of the architecture of the ANN node with the defined parameters to perform the computational task is checked. The ability to perform a computational task is determined on the basis of a comparison of the computational capabilities of the architecture and parameters of the ANN node and the necessary computing resources of the ANN node.

In the event of a discrepancy between the computing resources of the ANN node, the parameters of the ANN node are changed and in the event that it is impossible to increase the computing resources of the node, the architecture and parameters of the ANN node are changed.

Steps 8, 9. Checking the capabilities of the architecture to process the amount of information coming to its input.

Since each node ARANX-$N^{[l]}$ model is configured independently of others and is essentially a separate neuro-fuzzy system, to improve the quality of the obtained forecasts, it is possible to use the idea of combining an ensemble of neural networks [16].

This approach naturally leads to the architecture of the weighted ANARX neuro-fuzzy-WANARX system shown in **Fig. 4.9**.

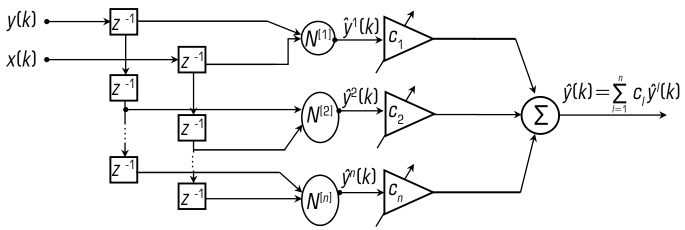

Fig. 4.9 Architecture of the WANARX system

At the same time, the output signal of such a system can be written in the form:

$$\hat{y}(k)=\sum_{l=1}^{n}c_l\hat{y}^l(k)=c^{\mathrm{T}}\hat{\vec{y}}(k), \tag{4.65}$$

where $\hat{\vec{y}}(k)=\left(\hat{\vec{y}}^1(k),\hat{\vec{y}}^2(k),\ldots,\hat{\vec{y}}^n(k)\right)^{\mathrm{T}}$, vector, weighting coefficients that can be adjusted and determine the proximity of signals $\hat{y}^l(k)$ – to the forecasted (processed) process and meets the non-displacement conditions:

$$\sum_{l=1}^{n}c_l=c^{\mathrm{T}}I_n=1, \tag{4.66}$$

where $I_n-(n\times 1)$ is a vector formed by units.

To find the vector c in batch mode, it is possible to use the method of uncertain Lagrange multipliers. For this, a sequence of errors is considered:

$$v(k)=y(k)-\hat{y}(k)=y(k)-c^{\mathrm{T}}\hat{\vec{y}}(k)=c^{\mathrm{T}}I_ny(k)-c^{\mathrm{T}}\vec{y}(k)=$$
$$=c^{\mathrm{T}}\left(I_ny(k)-\vec{y}(k)\right)=c^{\mathrm{T}}V(k). \tag{4.67}$$

Lagrange function:

$$L(c,\lambda)=\sum_k c^{\mathrm{T}}V(k)c+\lambda\left(c^{\mathrm{T}}I_n-1\right)=c^{\mathrm{T}}Rc+\lambda\left(c^{\mathrm{T}}I_n-1\right), \tag{4.68}$$

where λ is the undefined Lagrange multiplier; $R=\sum_k V(k)V^{\mathrm{T}}(k)$ is the error correlation matrix.

The Karush-Kun-Tucker system of equations is also considered:

$$\begin{cases}\nabla_c L(c,\lambda)=2Rc+\lambda I_n=\vec{0};\\ \dfrac{\partial L}{\partial\lambda}=c^{\mathrm{T}}I_n-1=0.\end{cases} \tag{4.69}$$

The solution of system (4.69) will lead to:

$$\begin{cases}c=R^{-1}I_n\left(I_n^{\mathrm{T}}R^{-1}I_n\right)^{-1};\\ \lambda=-2I_n^{\mathrm{T}}R^{-1}I_n,\end{cases} \tag{4.70}$$

at the same time, the Lagrangian (4.71) takes on the value at the saddle point:

$$L\cdot(c,\lambda)=\left(I_n^{\mathrm{T}}R^{-1}I_n\right)^{-1}. \tag{4.71}$$

The implementation of algorithm (4.69) may encounter significant difficulties while processing information online and with a high degree of signal correlation $\hat{y}^{l}(k)$. This leads to an ill-conditioned matrix R, which must be rotated at each real-time clock k.

Let's write the Lagrange function (4.67) in the form:

$$L(c,\lambda)=\sum_k y(k)-c^{\mathrm{T}}\hat{\vec{y}}(k)^2+\lambda\left(c^{\mathrm{T}}I_n-1\right), \tag{4.72}$$

and the gradient algorithm for finding its saddle point based on the Arrow-Hurwitz procedure [17, 18]:

$$\begin{cases}c(k)=c(k-1)-\eta_c(k)\nabla_c L(c,\lambda);\\ \lambda(k)=\lambda(k-1)+\eta_\lambda(k)\dfrac{\partial L(c,\lambda)}{\partial\lambda},\end{cases} \tag{4.73}$$

or

$$\begin{cases}c(k)=c(k-1)+\eta_c(k)\left(2\left(y(k)-c^{\mathrm{T}}(k-1)\hat{\vec{y}}(k)\right)\hat{\vec{y}}(k)-\lambda(k-1)I_n\right)=\\ =c(k-1)+\eta_c(k)\left(2v(k)\hat{\vec{y}}(k)-\lambda(k-1)I_n\right);\\ \lambda(k)=\lambda(k-1)+\eta_\lambda(k)\left(c^{\mathrm{T}}(k)I_n-1\right),\end{cases} \tag{4.74}$$

where $\eta_c(k)$, $\eta_\lambda(k)$ are the learning step parameters.

The Arrow-Hurwitz procedure reduces to a saddle point under fairly general assumptions of the relative values $\eta_c(k)$, $\eta_\lambda(k)$, however, to speed up the learning process, these parameters can be optimized. To do this, multiply the first relation (4.74) on the left by $\hat{\vec{y}}(k)$:

$$\hat{\vec{y}}(k)c(k)=\hat{\vec{y}}^{\mathrm{T}}c(k-1)+\eta_c(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right), \tag{4.75}$$

and introduce an additional function that characterizes the criterion convergence:

$$\begin{aligned}\left(y(k)-\hat{\vec{y}}^{\mathrm{T}}(k)c(k)\right)^2&=v^2(k)-2\eta_c(k)v(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right)+\\&+\eta_c^2(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right).\end{aligned} \tag{4.76}$$

The solution of the differential equation:

$$\begin{aligned}\frac{\partial\left(y(k)-\hat{\vec{y}}^{\mathrm{T}}(k)c(k)\right)^2}{\partial\eta_c(k)}&=-2v(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right)+\\&+2\eta_c(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right)^2=0,\end{aligned} \tag{4.77}$$

allows to obtain optimal values of the learning step $\eta_c(k)$ in the form:

$$\eta_c(k)=\frac{v(k)}{2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n}, \tag{4.78}$$

substituting which in the work (4.20), it is finally possible to write:

$$\begin{cases}c(k)=c(k-1)+\dfrac{v(k)\left(2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n\right)}{2v(k)\left\|\hat{\vec{y}}(k)\right\|^2-\lambda(k-1)\hat{\vec{y}}^{\mathrm{T}}I_n};\\ \lambda(k)=\lambda(k-1)+\eta_\lambda(k)\left(c^{\mathrm{T}}(k)I_n-1\right).\end{cases} \tag{4.79}$$

It is easy to see that the procedure (4.79) coincides with the Kachmaz-Widrow-Hoff algorithm (4.61).

A method of learning artificial neural networks for intelligent decision-making support systems is proposed. The operation of the proposed method was simulated in the MathCad 14 software environment.

To demonstrate the effectiveness of the proposed weighted ANARX system, a time series forecast was performed. To conduct the experiment, a training sample containing data on the

intelligence object was used. 5000 observations from this sample were used for experiments. The training sample contained 3000 observations, the test sample – 2000 observations. The square root of the root mean square error (RMSE) was used as a criterion for forecasting quality. Multilayer Perceptron (MLP), Radial Basis Neural Network (RBNN) and Adaptive Neuro Fuzzy Inference System (ANFIS) were used to compare the prediction quality.

The forecasting results for various systems are presented in the **Table 4.3**.

Table 4.3 The prediction of results for different systems

The system name	The number of configurable parameters	RMSE (research)	RMSE (test)	Time, s
Multilayer perceptron	51	0.1058	0.1407	0.1081
Radial-basis neural network	21	0.1066	0.2155	0.1081
ANFIS	80	0.0559	0.1965	0.1081
Evolving cascade system with neo-fuzzy nodes	20	0.0784	0.1081	0.1081

Since the multilayer perceptron cannot work online, 2 variants of this system were chosen for comparison.

The first multilayer perceptron was trained for one epoch (which is essentially similar to online data processing). The second multilayer perceptron was trained for five epochs and the number of adjustable parameters in this case was approximately equal to the number of adjustable parameters in the proposed systems. In both cases, multilayer perceptrons contained 4 inputs and 7 nodes in the hidden layer. The number of adjustable parameters was equal to 43. For the second multilayer perceptron, the operating time was almost 2 times longer, but, at the same time, the quality of prediction was also almost 2 times better. Two radial-basis neural networks were also selected.

The number of parameters of the first radial basis neural network was almost equal to the number of parameters of the proposed systems. The architecture of the second radial basis neural network was chosen taking into account the quality of its work. In the first case, the radial basis neural network had 3 inputs and 7 kernel functions. In the second case, the radial-basis neural network also had 3 inputs, but 12 kernel functions.

The ANFIS system showed one of the best prediction results in this experiment. It contained 4 inputs, 55 nodes and trained for five epochs. For training parameter α was set equal to 0.62. The system contained 37 adjustable parameters. *B*-splines with $q=2$ (triangular membership functions). The quality of the prediction of this system was quite high and the training time was the least.

The research of the developed method showed that the specified training method provides an average of 10–18 % higher efficiency of training artificial neural networks and does not accumulate errors during training (**Tables 4.3, 4.4**). This can be seen by the efficiency of data processing in the last columns of the **Tables 4.1, 4.2**.

Table 4.4 Comparison of forecasting results

Systems	The number of configurable parameters	RMSE (research)	RMSE (test)	Time, s
MLP (option 1)	43	0.0600	0.0700	0.4063
MLP (option 1)	43	0.0245	0.0381	0.9219
RBNN (option 1)	36	0.0681	0.0832	0.6562
RBNN (option 2)	61	0.0463	0.0604	1.0250
ANFIS	80	0.0237	0.0396	0.7031
ANARX with neuro-fuzzy nodes	37	0.0903	0.0922	0.4300
Weighted ANARX with neuro-fuzzy nodes	36	0.0427	0.0573	0.3750

CONCLUSIONS

1. Scientific novelty method of learning artificial neural networks, with an evolving architecture, consists of the following:

– it conducts training not only of the synaptic weights of the artificial neural network, but also of the type and parameters of the membership function;

– in case of impossibility to ensure the given quality of functioning of artificial neural networks due to parameter learning, the architecture of artificial neural networks is learned;

– the selection of the architecture, type and parameters of the membership function takes into account the computing resources of the tool and the type and amount of information entering the input of the artificial neural network;

– there is no accumulation of learning errors of artificial neural networks as a result of processing information received at the input of artificial neural networks.

2. The practical value of the proposed method lies in the fact that, based on it, practical recommendations were developed to increase the effectiveness of learning artificial neural networks.

The research of the developed method showed that the specified training method provides an average of 16–23 % higher efficiency of training artificial neural networks and does not accumulate errors during training.

3. Scientific novelties of Kohonen method of learning artificial neural networks with an evolving architecture are the following:

– it conducts training not only of the synaptic weights of the artificial neural network, but also of the type and parameters of the membership function;

– in case of impossibility to ensure the given quality of functioning of artificial neural networks due to parameter learning, the architecture of artificial neural networks is learned;

– data calculation takes place in one epoch without the need to store previous calculations;

– the selection of the architecture, type and parameters of the membership function takes into account the computing resources of the tool, the type and amount of information entering the input of the artificial neural network;

– there is no accumulation of learning errors of artificial neural networks as a result of processing information received at the input of artificial neural networks.

The research of the developed method showed that the specified training method provides an average of 10–18 % higher efficiency of training artificial neural networks and does not accumulate errors during training.

4. A method of learning cascaded artificial neural networks with an evolving architecture is proposed. The task of learning artificial neural networks has been formulated. It was established that the known methods of learning artificial neural networks have limited capabilities. It is proposed to use ANARX and WANARX artificial neural networks as a basis for developing the learning method.

5. An algorithm for learning the method of learning artificial neural networks for intelligent decision-making support systems has been developed.

Increasing the efficiency of information processing and reducing the assessment error is achieved due to:

– learning not only the synaptic weights of the artificial neural network, but also the type and parameters of the membership function;

– learning the architecture of artificial neural networks;

– the possibilities of combining elements of an artificial neural network;

– the opportunities to learn individual elements of an artificial neural network;

– the calculation of data for one epoch without the need to store previous calculations. This reduces the time for information processing due to the absence of the need to refer to the database;

– no lack of accumulation of learning errors of artificial neural networks as a result of processing information received at the input of artificial neural networks.

6. Conducted example of using the proposed method on the example of forecasting a time series of an intelligence object. The specified example showed an increase in the efficiency of the functioning of artificial neural networks at the level of 10–18 % in terms of the efficiency of information processing due to the use of additional training procedures of artificial neural networks.

REFERENCES

1. Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskyi, A., Yarosh, S., Gatsenko, S. et al. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (95)), 60–76. https://doi.org/10.15587/1729-4061.2018.144085
2. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The Method of Improving the Efficiency of Routes Selection in Networks of Connection with the Possibility of

Self-Organization. International Journal of Advanced Trends in Computer Science and Engineering, 8 (1.2), 1–6.
3. Zhang, J., Ding, W. (2017). Prediction of Air Pollutants Concentration Based on an Extreme Learning Machine: The Case of Hong Kong. International Journal of Environmental Research and Public Health, 14 (2), 114. https://doi.org/10.3390/ijerph14020114
4. Katranzhy, L., Podskrebko, O., Krasko, V. (2018). Modelling the dynamics of the adequacy of bank's regulatory capital. Baltic Journal of Economic Studies, 4 (1), 188–194. https://doi.org/10.30525/2256-0742/2018-4-1-188-194
5. Manea, E., Di Carlo, D., Depellegrin, D., Agardy, T., Gissi, E. (2019). Multidimensional assessment of supporting ecosystem services for marine spatial planning of the Adriatic Sea. Ecological Indicators, 101, 821–837. https://doi.org/10.1016/j.ecolind.2018.12.017
6. Çavdar, A. B., Ferhatosmanoğlu, N. (2018). Airline customer lifetime value estimation using data analytics supported by social network information. Journal of Air Transport Management, 67, 19–33. https://doi.org/10.1016/j.jairtraman.2017.10.007
7. Kachayeva, G. I., Mustafayev, A. G. (2018). The use of neural networks for the automatic analysis of electrocardiograms in diagnosis of cardiovascular diseases. Herald of Dagestan State Technical University. Technical Sciences, 45 (2), 114–124. https://doi.org/10.21822/2073-6185-2018-45-2-114-124
8. Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. Ice and Snow, 56 (4), 502–510. https://doi.org/10.15356/2076-6734-2016-4-502-510
9. Kanev, A., Nasteka, A., Bessonova, C., Nevmerzhitsky, D., Silaev, A., Efremov, A., Nikiforova, K. (2017). Anomaly detection in wireless sensor network of the "Smart Home" system. 2017 20th Conference of Open Innovations Association (FRUCT), 776 (20), 118–124. https://doi.org/10.23919/fruct.2017.8071301
10. Sreeshakthy M., Preethi J. (2016). Classification of Human Emotion from Deap EEG Signal Using Hybrid Improved Neural Networks with Cuckoo Search. Brain. Broad Research in Artificial Intelligence and Neuroscience, 6 (3–4), 60–73. Available at: https://lumenpublishing.com/journals/index.php/brain/article/view/1973
11. Chica, J., Zaputt, S., Encalada, J., Salamea, C., Montalvo, M. (2019). Objective assessment of skin repigmentation using a multilayer perceptron. Journal of Medical Signals & Sensors, 9 (2), 88–99. https://doi.org/10.4103/jmss.jmss_52_18
12. Massel, L. V., Gerget, O. M., Massel, A. G., Mamedov, T. G. (2019). The Use of Machine Learning in Situational Management in Relation to the Tasks of the Power Industry. EPJ Web of Conferences, 217, 01010. https://doi.org/10.1051/epjconf/201921701010
13. Abaci, K., Yamacli, V. (2019). Hybrid Artificial Neural Network by Using Differential Search Algorithm for Solving Power Flow Problem. Advances in Electrical and Computer Engineering, 19 (4), 57–64. https://doi.org/10.4316/aece.2019.04007
14. Mishchuk, O. S., Vitynskyi, P. B. (2018). Neural Network with Combined Approximation of the Surface of the Response. Research Bulletin of the National Technical University of

Ukraine "Kyiv Politechnic Institute", 2, 18–24. https://doi.org/10.20535/1810-0546.2018.2.129022

15. Kazemi, M., Faezirad, M. (2018). Efficiency estimation using nonlinear influences of time lags in DEA Using Artificial Neural Networks. Industrial Management Journal, 10 (1), 17–34. https://doi.org/10.22059/imj.2018.129192.1006898
16. Parapuram, G., Mokhtari, M., Ben Hmida, J. (2018). An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. Energies, 11 (3), 680. https://doi.org/10.3390/en11030680
17. Prokoptsev, N. G., Alekseenko, A. E., Kholodov, Y. A. (2018). Traffic flow speed prediction on transportation graph with convolutional neural networks. Computer Research and Modeling, 10 (3), 359–367. https://doi.org/10.20537/2076-7633-2018-10-3-359-367
18. Bodyanskiy, Y., Pliss, I., Vynokurova, O. (2013). Flexible Neo-fuzzy Neuron and Neuro-fuzzy Network for Monitoring Time Series Properties. Information Technology and Management Science, 16 (1). https://doi.org/10.2478/itms-2013-0007
19. Bodyanskiy, Ye., Pliss, I., Vynokurova, O. (2013). Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks. Oil and Gas Power Engineering, 2 (20), 158–162.
20. Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Upper Saddle River: Prentice Hall, Inc., 842.
21. Nelles, O. (2001). Nonlinear System Identification. Berlin: Springer, 785. https://doi.org/10.1007/978-3-662-04323-3
22. Wang, L.-X., Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Transactions on Neural Networks, 3 (5), 807–814. https://doi.org/10.1109/72.159070
23. Kohonen, T. (1995). Self-Organizing Maps. Berlin: Springer-Verlag, 362. https://doi.org/10.1007/978-3-642-97610-0
24. Kasabov, N. (2003). Evolving Connectionist Systems. London: Springer: Verlag, 307. https://doi.org/10.1007/978-1-4471-3740-5
25. Sugeno, M., Kang, G. T. (1988). Structure identification of fuzzy model. Fuzzy Sets and Systems, 28 (1), 15–33. https://doi.org/10.1016/0165-0114(88)90113-3
26. Ljung, L. (1987). System Identification: Theory for the User. Upper Saddle River: Prentice Hall, Inc., 432.
27. Otto, P., Bodyanskiy, Y., Kolodyazhniy, V. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering, 10 (4), 399–409. https://doi.org/10.3233/ica-2003-10409
28. Narendra, K. S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, 1 (1), 4–27. https://doi.org/10.1109/72.80202
29. Alieinykov, I., Thamer, K. A., Zhuravskyi, Y., Sova, O., Smirnova, N., Zhyvotovskyi, R. et al. (2019). Development of a method of fuzzy evaluation of information and analytical support of

strategic management. Eastern-European Journal of Enterprise Technologies, 6 (2 (102)), 16–27. https://doi.org/10.15587/1729-4061.2019.184394

30. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. https://doi.org/10.15587/1729-4061.2019.180197

DOI: 10.15587/978-617-8360-04-7.CH5

Svitlana Kashkevich, Illia Dmytriiev, Inna Shevchenko,
Oleksandr Lytvynenko, Lyubov Shabanova-Kushnarenko, Nataliia Apenko

CHAPTER 5

SCIENTIFIC-METHOD APPARATUS FOR IMPROVING THE EFFICIENCY OF INFORMATION PROCESSING USING ARTIFICIAL INTELLIGENCE

ABSTRACT

In this section of the research, a scientific and method apparatus for increasing the efficiency of information processing using artificial intelligence is proposed. The basis of this research is the theory of artificial intelligence, namely evolving artificial neural networks, basic genetic algorithm procedures and bio-inspired algorithms.

During the research, the authors proposed:

– the method of parametric optimization based on the improved wolf flock algorithm;

– the method of parametric evaluation of the control object based on the improved firefly algorithm;

– the method of finding solutions using the improved locust swarm algorithm;

– the method of finding solutions using the improved emperor penguin algorithm.

As a criterion for the efficiency of the specified scientific and method apparatus, the promptness of decision making regarding the parametric control of the state of the object with the given reliability was chosen. This makes it possible to create a hierarchical description of a complex process by levels of generalization and conduct an appropriate analysis of its state. The use of the proposed scientific and method apparatus will allow:

– to reduce the probability of premature convergence of the algorithm;

– to maintain a balance between the convergence speed of the algorithm and diversification;

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of search by flock agents;

– to carry out the initial display of individuals taking into account the type of uncertainty;

– to conduct accurate training of AS individuals;

– to conduct a local and global search taking into account the degree of noise of data on the state of the analysis object;

– to apply as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the local extremum problem.

KEYWORDS

Bio-inspired algorithms, multi-agent systems, wolf flock algorithm, reliability and adequacy.

5.1 THE METHOD OF PARAMETRIC OPTIMIZATION BASED ON THE IMPROVED WOLF FLOCK ALGORITHM

Computational intelligence methods have become widespread for solving a variety of complex tasks, both purely scientific and in the field of technology, business, finance, medical and technical diagnostics, and other fields. These include intelligent data analysis (Data Mining), dynamic data analysis (Dynamic Data Mining), analysis of data streams (Data Stream Mining), analysis of large arrays of information (Big Data Mining), Web-Mining, Text Mining [1–6].

The increase in volumes of information circulating in various systems of information collection, processing and transmission leads to significant use of computing resources of hardware. The armed forces of technically developed countries have integrated decision-making architectures based on the works [7–15]:

– artificial intelligence and nanotechnologies;

– effective processing of large amounts of information;

– data compression technologies to increase the speed of their processing.

At the same time, the use of information systems with elements of artificial intelligence will make it possible to increase the efficiency of planning, conducting operations (combat operations) and their comprehensive support, will affect the doctrine, organization and methods of application of groups of troops (forces).

At the same time, increasing the dynamism of operations (combats), increasing the number of various sensors and the need to integrate them into a single information space creates a number of problems:

– implemented algorithms for establishing correlations between events do not fully take into account the reliability of sources of intelligence information and the reliability of information in the dynamics of operations (combats);

– forms of information presentation complicate its transmission through communication channels;

– limited computing power of hardware;

– radio electronic suppression of SW and USW radio communication channels and cybernetic influence on information systems;

– transition to the principle of monitoring objects assessment "everything affects everything at once", which covers aggregate network and computing resources of all types of armed forces.

That is why it is necessary to develop algorithms (methods and techniques) that are capable of solving optimization problems from various sources of intelligence in a limited time and with a high degree of reliability.

The problem that needs to be solved in the research is to increase the efficiency of solving optimization problems while ensuring the given reliability.

Considering the above, an actual scientific task is an improvement of the optimization method based on the wolf flock algorithm, which would allow to increase the efficiency of the decisions made regarding the management of the parameters of the control object with the specified reliability.

Wolves have a typical family lifestyle: they live in flock – family groups consisting of a pair of leaders, their relatives and lone wolves. A strictly defined hierarchy is observed within the flock, at the top of which is the flock leader, who directs other individuals to search for prey. Wolves "explore" the area for the presence of a victim, when one of them smells the victim, the search for it begins. The stronger the smell, the closer the wolves are to the prey. Thus, they move in the direction of increasing the smell of the victim.

The "wolf flock" search method copies the process of their hunting. Suppose that the area on which wolves hunt is a search area in the sense of optimization and the flock is wolves. At the same time, these procedures are not without some shortcomings that worsen the properties of the global extremum search process.

An analysis of works [9–37] showed that the common shortcomings of the above-mentioned researches are:

– the lack of possibility of forming a hierarchical system of indicators;

– the lack of consideration of computing resources of the system;

– the lack of mechanisms for adjusting the system of indicators during the assessment;

– the lack of consideration of the type of uncertainty about the management object state, which creates corresponding errors in the assessment of its real state;

– the lack of deep learning mechanisms of knowledge bases;

– high computational complexity;

– the lack of consideration of computing (hardware) resources available in the system;

– the lack of search priority in a certain direction.

For this purpose, it is proposed to improve the method of parametric optimization based on the improved wolf flock algorithm.

The aim of research is the improvement method of parametric optimization based on the improved wolf flock algorithm. This will allow to increase the efficiency of optimization with the given reliability and the development of subsequent management decisions. This will make it possible to develop software for intelligent decision-making support systems in the interests of the combat management of the actions of troops (forces).

To achieve the aim, the following tasks were set:

– to carry out a mathematical formulation of the research task;

– to determine the method implementation algorithm;

– to lead an example of the application of the proposed method in the analysis of the operational situation of a group of troops (forces).

Given: $I=\{1,\ldots,n\}$ is a set of points, matrix (c_{ij}) is the pairwise distances between points $1\le i, j\le n$.

Find: a contour (path) of minimum length, thus, a cycle that passes through each vertex exactly once and has minimum weight.

Let's carry out a mathematical formulation of the parametric optimization task using the wolf flock algorithm:

$$\text{Variable tasks: } x_{ij}=\begin{cases}1, \text{ if there is a way;}\\ 0, \text{ otherwise.}\end{cases}$$

Find the objective function of the form:

$$J(x)=\min\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}x_{ij}, \tag{5.1}$$

where c_{ij} is the distance between points i and j under the following restrictions:

$$\sum_{j=1}^{N}x_{ij}=1,\ \forall j=\overline{1,N},\quad \sum_{i=1}^{N}x_{ij}=1,\ \forall i=\overline{1,N}. \tag{5.2}$$

Let N "wolves" be generated in the Euclidean space of dimension D, thus, each wolf is represented in the form of a vector $x_i(x_{i1},\ldots,x_{iD})$, which determines its coordinates in space. Thus, the swarm (population) represents a set of potential solutions, the coordinates of which, just as for the swarm optimization algorithm [2], are updated at each iteration until the optimal solution is found or the maximum set number of calculations of the objective function is performed.

Then the function $f(x)$, which characterizes how strongly the smell of the victim is felt by wolves, is the target, and the coordinates of the victim itself are the optimal point. The distance between two wolves p and q is described by the species metric: $L(p, q)$.

The "wolf flock" algorithm searches for the optimal victim point. They divide into groups, move in different directions and exchange information among themselves.

The optimization algorithm based on a wolf flock consists of the following sequence of actions:

Step 1. Input of output data.

At this stage, the initial data available in the system are entered to solve the optimization problem.

Step 2. Placing wolves in initial positions taking into account uncertainty.

Create a wolf flock in the form of a set of Euclidean vectors distributed over a set of admissible values of the arguments, taking into account the degree of χ awareness of the object state.

Division of the degree of uncertainty: (full uncertainty – exposure randomly, partial uncertainty – exposure taking into account the coefficient of correction of the position of wolves $\chi = 0.01 \div 0.99$).

Step 3. Determination of leaders in the flock.

In the classic wolf flock algorithm, the "wolf" with the best value of the objective function at this iteration is the leader. If at the next iteration another "individual" is found with a better value of the objective function than that of the leader, then, accordingly, the flock "finds" a new leader. In the specified procedure, it is proposed to determine the number of leaders that will ensure the maximization of the efficiency of the search with restrictions on the available computing resources.

Step 4. Search for prey by other wolves in the flock.

Other wolves explore the area for prey. Moreover, the function $f(x_i)$ characterizes how strongly the smell of the victim is felt by the *i*-th "wolf". Then the value of G_{best} characterizes how strongly the smell of the victim is felt by the flock leader.

Step 5. Change of the flock leader.

If $f\left(x_i\right) > G_{Best}$, then the *i*-th "wolf" is closer to the victim than the leader of the flock, so the *i*-th wolf becomes the leader at this stage $f\left(x_i\right) = G_{Best}$. If $f\left(x_i\right) < G_{Best}$, then the "wolf" moves in space with some predetermined *step*.

The leader(s) of the flock "inform" the other "wolves" in the flock about their location, as the closest point to the victim now, so that they move in its direction.

Step 6. Approaching the flock leaders.

At this stage, the "leader" ("leaders") is considered almost the same as the victim – a goal to which it is necessary to approach. Then the "wolves" of the flock move in the direction of the leader with a predetermined *step* and the coordinate *d* *i*-th "wolf" on the $(k+1)$-th iteration is calculated by the formula:

$$x_{iD}^{(k+1)} = x_{iD}^{(k)} + step\frac{G_{Best}^{(k)} - x_{iD}^{(k)}}{\left\| G_{Best}^{(k)} - x_{iD}^{(k)} \right\|}. \tag{5.3}$$

From formula (5.3) and the description of the algorithm, it can be seen that in the "wolf flock" search method, only the coordinates of the "wolves" are updated without taking into account the speed of their movement in space. Four parameters must be selected for the swarm algorithm (learning coefficients, inertial weight, population size). However, for the wolf flock search method, it is enough to select only two parameters – the population size *N* and the *step*, from which the "wolves" move in the direction of the leader and the victim.

It should be noted that formula (5.3) cannot be applied to the traveling salesman problem in a standard way. In this case, only its main principle is taken, namely: other "wolves" must be sufficiently "similar" to their leader(s), who in the current iteration is closer to the victim (by the value of the objective function). At this stage, using the improved genetic algorithm proposed by the authors in the work [24].

Let there be a population and a corresponding adaptation of each wolf. An example of the algorithm is shown in **Tables 5.1–5.3**.

Table 5.1 An example of "wolf chromosomes" with the value of adaptability

Chromosomes (the order of passing points by a traveling salesman)						The value of the fitness function (passed by the corresponding "wolf")
3	1	6	4	2	5	0.31
5	6	2	1	3	4	0.32
3	4	2	5	1	6	0.021
4	1	6	2	5	3	0.32

Then, it is possible to determine the best "wolf" by its value of the fitness function (**Table 5.2**).

Table 5.2 The best chromosome of the population

The best chromosome						The value of fitness functions
3	4	2	5	1	6	0.0263

Based on the best chromosome (**Table 5.2**), it is possible to generate a new population based on the work [24] (**Table 5.3**).

Table 5.3 Generation of a new population based on the wolf with the best adaptation (by the length of the path traveled by it)

Chromosomes (order of passing points)						The value of the fitness function (the path traveled by the corresponding wolf)
3	4	2	5	1	6	0.022
2	3	4	5	1	6	0.19
3	4	2	5	6	1	0.04
1	4	2	5	6	3	0.21

All subsequent iterations are performed in a similar way: the best "wolf" is found and a new population is generated based on it.

The end of algorithm.

The comparative analysis of the obtained work results was carried out on the basis of two criteria: the criterion of time and the criterion of the optimal distance traveled by the traveling salesman found by each algorithm for a different number of points (from 30 to 300). Tabular results are given below (**Tables 5.4, 5.5**).

Table 5.4 Comparative analysis of the classic and modified "wolf flock" algorithm for the traveling salesman problem based on the distance objective function criterion

The number of vertices	Population size	Maximum number of iterations	Classic wolf flock algorithm		Modified wolf flock algorithm		Accurate solution to a minimum
			f_{Best}	Error in %	f_{Best}	Error in %	
30	30	1000	23.95767	1.2	21.4	0.00	23.584849
50	60	5000	429.5757	1.25	416.2	0.1	421.787667
100	100	10000	534.5848	1.7	520.7	1.16	523.584849
150	200	20000	312.7	2.28	308.2	1.22	328.087454
300	500	50000	881.8	3.75	846.5	1.5	854.154940

As can be seen from the **Table 5.4** that the modified wolf flock algorithm gives more accurate results in contrast to the classical one, while the accuracy increases to 30 %.

Table 5.5 Comparative analysis of the classic and modified wolf flock algorithm for the traveling salesman problem according to the criterion of working time

The number of vertices	Population size	Maximum number of iterations	Algorithm operation time (in seconds)	
			Classic wolf flock algorithm	Modified algorithm of wolves flock
30	30	1000	2.15978	1.8
50	60	5000	10.79888	11.6
100	100	10000	53.99440	50.1
150	200	20000	269.97198	270.3
300	500	50000	1349.85991	1355.24

According to the calculation results given in **Table 5.5**, it is established that the operating time of the modified algorithm is 10 % longer. This is due to the fact that the entire population is divided into subgroups, each of which has its own "leaders".

Let's compare how effective the modification of this algorithm is compared to population algorithms, such as "particle swarm" optimization and the classic genetic algorithm, which is most often used to solve the traveling salesman problem (**Tables 5.6, 5.7**).

As it is possible to see, the modified wolf flock algorithm here also gives much more accurate results, unlike the other two considered (up to 50 % on average).

According to the results given in the **Table 5.5**, it can be seen that the operating time of the modified algorithm is on average 20 % less than that of the "particle swarm" algorithm from those given in the **Table 5.7**. The main time of work in the genetic algorithm is allocated to crossbreeding and obtaining pairs of chromosomes.

Table 5.6 Comparative analysis of the modified "wolf flock" algorithm with the genetic algorithm and the particle swarm algorithm according to the optimal path criterion

The number of vertices	Population size	The classic swarm particles algorithm		Classic genetic algorithm		Modified wolf flock algorithm		Exact solution to a minimum
		f_{Best}	Error in %	f_{Best}	Error in %	f_{Best}	Error in %	
30	30	24.78769	5.10	24.48957	3.84	23.12	0.00	23.584849
50	60	431.75587	2.36	428.76556	1.65	411.9	0.11	421.787667
100	100	538.56887	2.86	538.85479	2.92	524.4	1.2	523.584849
150	200	338.66566	3.22	334.77659	2.04	312.4	1.36	328.087454
300	500	902.66575	5.68	887.56746	3.91	850.6	1.7	854.154940

Table 5.7 Comparative analysis of the modified wolf flock algorithm with the genetic algorithm and the "particle swarm" algorithm according to the criterion of operation time

The number of vertices	Population size	Maximum number of iterations	Algorithm operation time (in seconds)		
			Classic particles swarm algorithm	Classical genetic algorithm	Modified wolf flock algorithm
30	30	1000	2.74546	2.43579	1.33
50	60	5000	11.64576	18.85740	11.83
100	100	10000	53.65869	61.27783	50.7
150	200	20000	269.76457	271.19442	260.1
300	500	50000	1349.57659	1356.37387	1298.7

The obtained results on increasing the efficiency of the optimization are explained by the use of the improved wolf flock algorithm in contrast to the classical empirical expressions and the classical wolf flock algorithm. The wolf flock algorithm is not used in its classical form, but by improving it with the help of improved procedures for exhibiting wolves, taking into account the type of uncertainty and using additional procedures developed by the authors in the work [24].

5.2 THE METHOD OF PARAMETRIC ASSESSMENT OF THE CONTROL OBJECT BASED ON THE IMPROVED FIREFLY ALGORITHM

A mathematical model of any object is a formalized description of a quality criterion that ensures the fulfillment of specified functions, requirements, etc. [6, 7].

In order to further develop the method of parametric assessment of the management object, it is proposed to develop a mathematical model of simulating a colony of firefly.

During the development of the algorithm model, the life of a swarm of firefly was not simulated, which unambiguously copied the existing natural ecosystem, but a colony simulation was used as an optimization device, in which the system is slightly different from the natural one.

Therefore, for the formal description of the model, it is possible to use the concept of "agent" instead of the concept of "firefly". While initializing the search, all agents are randomly distributed in the search space of the objective function.

Each agent releases a certain amount of luciferin and has its own decision-making area. Agent *i* considers another agent *j* as a neighbor if it is within the radius of agent *i* search neighborhood and agent *j* luciferin level is higher than agent *i*, so $l_j > l_i$. The local decision-making area is determined by the radius of the search neighborhood for each *i*-th agent.

Using a probabilistic mechanism, each agent selects a neighboring agent with a higher luciferin level than its own and moves in its direction. In other words, each agent moves in the direction of the agent with a higher glow level.

The glow intensity of each agent is determined by the value of the objective function at the current position. The higher the intensity of the glow, the greater the value of the target function [13]. In addition, the radius of each agent's search neighborhood depends on the number of agents in that search area. If there is a small number of agents in the search neighborhood, its radius increases. Otherwise, the search radius is reduced. This algorithm has the following global stages: initial distribution of agents in the search space, updating the luciferin level, moving agents to a more promising search area, updating the search radius of each agent [14–16].

The task of parametric optimization of the state of the control object is to find such internal parameters of the construction, in which the initial parameters would have the given characteristics and the construction elements and the method of their connection would remain unchanged.

Let the control object have *n* controlled parameters forming the vector $X=(x_1, x_2, \ldots, x_n)$. Let's define the objective function by $F(X)$ and the domain of its definition by *XO*. Vector *X* determines the coordinates of a point in the *XO* definition area. If the elements of the vector *X* take only discrete values, then the *XO* is a discrete set of points and the optimization task belongs to the field of discrete programming.

The purpose of the algorithms for solving the parametric optimization problem is to determine such a vector of control parameters (influences) in which the given objective function acquires a minimum value.

In the process of developing a mathematical model, it is necessary to determine the parameters of the object that affect the criterion of optimality. Then, the parametric, discretization and functional restrictions imposed on the parameters of the control object are determined [5, 6].

Constraints of the following type are called parametric:

$$x_i' \le x_i \le x_i'', \tag{5.4}$$

where x_i is the *i*-th parameter of the object; x_i' and x_i'' are respectively min and max values of the *i*-th parameter.

Discretization constraints have the form:

$$x_j = \left\{ x_{j1}, x_{j2}, \ldots, x_{jm} \right\}, \tag{5.5}$$

where x_j is the j-th parameter of the control object; x_{jk} is the permissible values of the j-th parameter ($k = 1,2,\ldots, m$). Such restrictions are imposed on the values of the parameters or in connection with their physical nature.

Functional restrictions imposed on the parameters of control objects are conditions for the connection of their values. These restrictions are:

$$g_i(x) \le 0;\ g_j(x) = 0;\ g_k(x) < 0. \tag{5.6}$$

Functional limitations in the optimal management of the object can be the conditions of fault protection of the control system, survivability and mobility. These conditions ensure the desired values of certain technical characteristics [7–9].

From the expressions (5.1)–(5.3), it can be concluded that the expressions allow to describe the processes in the object of control, to determine the controlled parameters of the object and also to describe the cause-and-effect relationships between them. The specified description is universal and allows to describe the management object taking into account the hierarchy and individual specifics of each management object.

The method of parametric management of the state of the control object consists of the following sequence of actions:

Step 1. Initialization of input parameters. At this stage, initial parameters are entered for calculating the state of the management object and justifying subsequent management decisions.

Step 2. Placement of the initial (current) population. The initial placement of agents takes place taking into account the type of uncertainty about the state of the control object. Unlike the classic firefly algorithm, the initial placement takes into account the type of uncertainty about the state of the control object and the degree of data noise. The corresponding correction coefficients are given in the work [2]. Placement in the search space of the initial (current) population of solutions in the parametric control problem consisting of n agents.

Initially, all agents have the same amount of luciferin. At each iteration, the level of luciferin is updated and then the position of the agent in space is replaced based on the given rules. The state of the agent s_i, $\in i[l:|S|]$ is determined by the following variables: X_i is the current state of the agent in the search space; l_i is the agent glow level; r_i is the radius of the neighborhood.

Agent s_i is considered a neighbor of agent $s_j \in i$, $j[l:|S|]$, $i \neq j$, if the following conditions are met: the Euclidean distance between agents does not exceed the current radius r_i; the current glow level of agent s_j exceeds the same level of agent s_i, $\in i[l:|S|]$. If the agent has several neighbors, then the agent randomly chooses one of them with a probability that is proportional to their luminance levels (roulette rule).

CHAPTER 5

Let's suppose that agent s_i chose agent s_j.

Then the new position of agent s_i is determined by the formula:

$$X_i^{'} = X_i + \lambda \frac{X_j - X_i}{\left\| X_j - X_i \right\|}, \; i, j\left[l : \left| S \right| \right], \; i \neq j, \tag{5.7}$$

where λ is a constant value of the step (a free parameter of the algorithm).

The new radius of the neighborhood of the agent s_i is determined according to the expression:

$$r_i' = \min\left(r_{\min}, \max\left(0, r_i + \varepsilon\left(n - \left| N_i \right| \right) \right) \right), \; i \in \left[1 : \left| S \right| \right], \tag{5.8}$$

where N_i is the current set of neighbors of agent s_i; $r_{\min}$ is the minimum permissible radius of the neighborhood; n is the desired number of neighbors.

Step 3. The update of the luciferin level depends on the position of the agent in space (the value of its objective function). All agents at the initial iteration have the same level of luciferin, so the value of the objective function of each agent depends on its position in the search space.

The level of luciferin of each agent increases in proportion to the measured characteristics of the agent (temperature, radiation level). From the point of view of optimization, this is the objective function. To simulate the process of decay of a fluorescent substance, a part of luciferin is subtracted.

The calculation of the luciferin level $l_i(t)$ (luminescence level) of the i-th agent at time t is shown below:

$$l_j\left(t + 1 \right) = \left(1 - \rho \right) l_j\left(t \right) + \gamma J_j\left(t + 1 \right), \tag{5.9}$$

where ρ is the attenuation coefficient of the luciferin level for modeling the decay process of the fluorescent substance ($0 < \rho < 1$); γ is the attractiveness coefficient of the agent; J_j is the value of the objective function of the j-th agent at time t.

Step 4. Each agent selects the agent within the search neighborhood radius r_i, whose luciferin level is higher than its own. Task $N_i(t)$ is the set of neighbors of the i-th agent at time t, r_i is the radius of the search neighborhood of the i-th agent at time t.

Step 5. While updating its position in the search space, each agent, based on a probabilistic mechanism, moves in the direction of the agent whose luciferin level is higher than its own.

For each i-th agent, the possibility of movement in the direction of agent j is determined by the formula:

$$p_{ij}\left(t \right) = \frac{l_j\left(t \right) - l_i\left(t \right)}{\sum\limits_{k \in N_i\left(t \right)} l_k\left(t \right) - l_i\left(t \right)}, \tag{5.10}$$

where $j \in N_i(t)$, $N_i(t) = \{j : d_{ij}(t) < r_d^i(t)\}$; $l_i(t) < l_j(t)$, $d_{ij}(t)$ is the Euclidean distance between agents i and j at time t; $l_j(t)$ is the luciferin level of agent j at time t; $r_d^i(t)$ is the local area of decision making of agent i that changes at time t.

Step 6. Agent i, using the roulette wheel method, selects agent j and moves in its direction. Then, the updated position of agent i is determined by the formula:

$$x_i(t+1) = x_i(t) + st \cdot \left\{ \frac{x_j(t) - x_i(t)}{\left\| x_j(t) - x_i(t) \right\|} \right\}, \tag{5.11}$$

where st is the step size.

Step 7. Update the radius of the search neighborhood r_d^i by the formula:

$$r_d^i(t+1) = \min\left\{ r_s, \max\left\{ 0, r_d^i(t) + \beta\left(n_t - \left|N_i(t)\right|\right) \right\} \right\}, \tag{5.12}$$

where β is a constant parameter and n_t is a parameter to control the number of neighboring agents.

Values ρ, γ, st, β, n_t are algorithm parameters, the value of which is determined experimentally.

Step 8. Training agents. At this stage, agents are trained using the method of learning artificial evolving neural networks proposed in the WPE [2].

Step 9. Determination of the need to attract additional computing resources. At this stage, with the help of the proposed work [36] approach calculates the necessary amount of computing resources that must be involved for assessment and parametric control of the control object state.

The end of algorithm.

Modeling was carried out to assess the state of the control object and subsequent parametric control. The operational grouping of troops (forces) was considered as an object of assessment and management. An operational grouping of troops (forces) formed on the basis of an operational command with a typical composition of forces and devices according to the wartime staff and with a range of responsibility in accordance with current regulations.

The simulation results made it possible to determine the dependence of the algorithm's operation time on the input parameters. The graph of the dependence of the time of the algorithm on the amount of input data is presented in **Fig. 5.1**.

The time complexity of the algorithm was $O(n^2)$, where n is the number of input parameters. The dependence of the algorithm's running time on the number of iterations was also considered. The simulation results are shown in **Fig. 5.2**.

This dependence is equal to $O(n^4)$, where n is the number of iterations.

To determine the effectiveness of the proposed method, researches were conducted in comparison with other swarm methods, namely the ant colony algorithm (ACO) and the particle swarm optimization (PSO) method. The results of the experiments are given in **Table 5.8**.

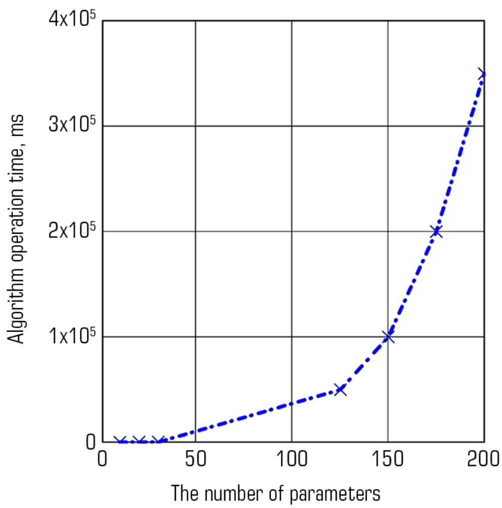

Fig. 5.1 Time complexity of the algorithm depending on the number of input parameters

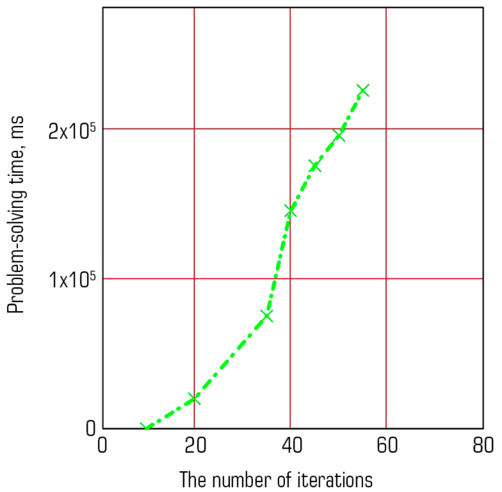

Fig. 5.2 Dependence of the problem-solving time on the number of iterations

It can be seen from the figures that the developed method outperforms the ACO and PSO algorithms in terms of the quality of the obtained solutions. In addition, the method operates with a smaller number of parameters and, accordingly, does not require large computational costs.

The main advantage of the method based on the behavior of a swarm of firefly is that while using it, the probability of hitting a local optimum is sharply reduced, and due to parallelization, the time is reduced. At each iteration, it is equal to the search time in the most promising block.

Table 5.8 Comparative analysis of swarm algorithms

The number of intermediate solution points	Particle swarm method	Ant algorithm	Proposed method
N	T, s	T, s	T, s
5	0.282	0.276	0.232
10	0.723	0.4	0.423
15	6.641	0.999	1.1
20	10.7	2.5	2.7
30	21.3	4.5	4.7
40	42	7.9	7.4
50	56	10.1	9.2
100	120	17.6	19.6
200	727	74.2	80.2
Complexity	$O\left(\frac{(N-1)!}{4}\right)=O(N!)$	$O(N^2+N)=O(N^2)$	$O(n^2)$

According to the results of the analysis of the data given in the **Table 5.8** it is possible to show that the proposed method has an acceptable computational complexity.

In the range (from 50 to 100), the proposed method becomes more efficient in terms of algorithm operation time compared to other algorithms (faster than the particle swarm method by 72.9–81.6 % and the ant method by 7–9.1 %. The proposed method allows to obtain adequate solutions with a complex hierarchical structure of the monitoring object, the effectiveness of the proposed method is on average from 17 to 20 % with different hierarchies of the construction of the control object.

The obtained results on increasing the efficiency of parametric control are explained by the use of an improved firefly algorithm in contrast to classical empirical expressions. The firefly algorithm is not used in its classical form, but by improving it with the help of evolving artificial neural networks.

5.3 THE METHOD OF FINDING SOLUTIONS USING THE IMPROVED LOCUST SWARM ALGORITHM

The proposed algorithm is an improved locust swarm algorithm and consists of the following sequence of actions:

Step 1. Input of initial data. At this stage, the initial data available about the object to be analyzed are entered. The existing model of the analysis object is also initialized. At this stage, the solution matrix D is filled: each column is filled with a subset of ω_i.

Step 2. Processing of initial data taking into account the degree of uncertainty.

At this stage, the type of uncertainty is taken into account about the object to be analyzed and initialization of the basic state model of the object to be analyzed [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. This is done with the help of correction coefficients.

Step 3. Initial exposure of the AC in the search area.

Let's describe the change in the position of an individual locust with individual behavior. Let x_i^k is the current position of the i-th AC in a swarm of N individuals. Then a new position x_i^{k+1} AC is calculated according to the following formula:

$$x_i^{k+1} = x_i^k + \Delta x_i, \tag{5.13}$$

where Δx_i is the change in the position of the i-th AC on the $(k+1)$-th iteration due to interaction with other AC of the swarm.

Two AC with individual behavior do not tend to get closer if there is a small distance between them and, on the contrary, they get closer, maintaining the cohesion of the swarm, if there is a significant distance between them. The force of attraction/repulsion of the AC is defined as the difference:

$$s(r) = kar \cdot e^{-\frac{r}{\lim}} - e^{-r}, \tag{5.14}$$

where r is the distance between a pair of AC; kar is the coefficient of attraction/repulsion of AC; lim is the permissible value of the distance between AC. If $kar < 1$ and $\lim > 1$, then this means that there is a small distance between the AC and repulsion is stronger than attraction. Then the force of influence of the j-th locust on the i-th locust is determined as follows:

$$s_{ij} = s(r_{ij}) \cdot d_{ij}, \tag{5.15}$$

where $r_{ij} = \left| x_j - x_i \right|$ is the distance between the j-th and i-th AC swarm; $d_{ij} = \left(x_j - x_i \right) / r_{ij}$ is a unit vector. Then the total attraction/repulsion force of the swarm for the i-th locust is defined as the superposition of all pairwise interactions:

$$S_i = \sum_{j=1, j \neq 1}^{N} s_{ij}. \tag{5.16}$$

Changing the position of the i-th locust Δx_i corresponds to the work (5.17):

$$\Delta x_i = S_i. \tag{5.17}$$

In contrast to individual behavior in swarm behavior, AC rapidly concentrates around individuals that have found food sources. In order to simulate equal behavior, is introduced for each AC x_i food index f_i $\left(f_i \in \left[0,1 \right] \right)$. Then, N individuals of the population are ranked according to the decrease of this index and then b individuals ($b \ll N$) with the highest food indicators are selected among them. Around each of the b individuals in a radius R_c a subset of locusts is randomly concentrated.

It is possible to assume that the entire search space is a plantation where the AC interact with each other. Each solution in the search space represents the position of the AC on the plantation and is characterized by the value of the fitness function, which reflects the level of the food index. The algorithm implements patterns of individual and swarm behavior, which are controlled by a set of operators that simulate these behavior patterns.

Population $L^k \left(\{ l_1^k, l_2^k, \ldots, l_N^k \} \right)$ with N individuals, the AC evolves from the initial position ($k=0$) to a given number of generations ($k=gen$). Every locust l_i^k $\left(i = 1 \ldots N \right)$ is an n-dimensional vector $\{ l_{i1}^k, l_{i2}^k, \ldots, l_{in}^k \}$, in which each element corresponds to a variable solution of the optimization problem.

The set of variable solutions makes up the admissible search space:

$$S = \left\{ l^k \in R^n \mid lb_d \leq l^k \leq ub_d \right\}, \tag{5.18}$$

where lb_d and ub_d correspond to the lower and upper bounds of dimension d, respectively. The food index level associated with each locust is estimated using the function $f_i \left(l_i^k \right)$.

In the *SIBL* algorithm, two behavior operators are used at each iteration of the evolution process: ***A*** – individual and ***B*** – swarm. Operator ***A*** is used to diversify the solution search space and operator ***B*** is used to refine the solution in a certain area of space. Let's consider each of the operators in more detail.

Operator ***A***, implementing the pattern of individual behavior of the AC, changes the current position l_i^k i-th ($i=1 \ldots N$) with speed v_i AC by magnitude Δl_i^k: $p_i = l_i^k + \Delta l_i^k$ taking into account the value of the fitness function and the position of the dominant AC of the swarm.

Step 4. Determination of the initial speed of the AC movement.

Initial speed v_0 of each AC is determined by the following expression:

$$v_i = \left(v_1, v_2 \ldots v_S\right), v_i = v_0. \tag{5.19}$$

Step 5. Generation of the search vector for each AC taking into account the degree of uncertainty:

$$\omega_i = \left(\left(\omega_{i1} \times \eta_{i1}\right), \left(\omega_{i2} \times \eta_{i2}\right), \ldots, \left(\omega_{in} \times \eta_{in}\right)\right). \tag{5.20}$$

To start the motion process, a vector of the direction of motion of the AC is generated:

$$\Delta\omega_i = \left(\Delta\omega_{i1}, \Delta\omega_{i2}, \ldots, \Delta\omega_{in}\right), \tag{5.21}$$

$$\Delta\omega_{ij} = \begin{cases} a, \text{ if } t = rand(0,1) > 1/2; \\ -a, \text{ if } t \le 1/2, \end{cases} \tag{5.22}$$

where $j = 1,2,\ldots,n$, a $(a > 0)$ is the step length and is chosen depending on the studied area.

The force of attraction/repulsion between the j-th and i-th individuals is calculated as follows:

$$s_{ij}^m = \rho\left(I_i^k, I_j^k\right) \cdot s\left(r_{ij}\right) \cdot d_{ij} + rand(1,-1), \tag{5.23}$$

where determined accordingly (1); $d_{ij} = \left(I_j^k - I_i^k\right)/r_{ij}$ is a unit vector directed from I_i^k to I_j^k; $rand(1,-1)$ is a random number from the interval (–1, 1); $\rho\left(I_i^k, I_j^k\right)$ is the dominance function between the j-th and i-th AC. To determine ρ, all AC of the population $L^k\left(\{I_1^k, I_2^k, \ldots, I_N^k\}\right)$ are ranked in decreasing order of their fitness functions. The best AC is assigned a rank of 0, the worst individual receives a rank of N–1. Thus, the function $\rho\left(I_i^k, I_j^k\right)$ is defined as follows:

$$\rho\left(I_i^k, I_j^k\right) = \begin{cases} e^{-\left(5 \cdot rank\left(I_i^k\right)/N\right)}, \text{ if } rank\left(I_i^k\right) < rank\left(I_j^k\right); \\ e^{-\left(5 \cdot rank\left(I_i^k\right)/N\right)}, \text{ if } rank\left(I_i^k\right) > rank\left(I_j^k\right), \end{cases} \tag{5.24}$$

where the $rank(\alpha)$ function indicates the rank of an individual. According to (5.24), the function ρ acquires values from the interval (0, 1) and the value 1 is reached when one of the individuals is the best element of the population and a value close to 0 – when both individuals have low indicators of the fitness function.

Finally, the total force of attraction/repulsion acting on the i-th individual is calculated as the superposition of all pairwise interactions:

$$S_i^m = \sum_{j=1, j\neq 1}^{N} s_{ij}^m. \tag{5.25}$$

Step 6. Calculation of the change in the value of the AC fitness function.

After calculating the new positions $P(\{p_1,p_2,\ldots,p_N\})$ of AC population L^k, it is necessary to change the values of the fitness functions $F(\{f_1,f_2,\ldots,f_N\})$. At the same time, only those changes that guarantee improvement of the search results are allowed. In other words, if $f_i(p_i) > f_i(l_i^k)$, then a new position is taken p_i, otherwise the position is maintained l_i^k:

$$f_i = \begin{cases} p_i, \text{ if } f(p_i) > f(l_i^k); \\ l_i^k, \text{ otherwise.} \end{cases} \tag{5.26}$$

Operator ***B***, which implements the AC swarm behavior pattern, is aimed at refining the solution in a certain area of the search space. To perform it, first, the AC suitability functions are sorted in descending order. Sorting results are stored in a set $B(\{b_1,b_2,\ldots,b_N\})$. Among them, the g best AC with the highest value of the fitness function stand out. They form a subset ***E*** of the most promising solutions. Around each individual with $f_i \in \boldsymbol{E}$, a subspace is created with radius $\boldsymbol{C}_j$, which is determined as follows:

$$e_d = \frac{\sum_{q=1}^{n}\left(ub_q - lb_q\right)}{n}\cdot\beta, \tag{5.27}$$

where ub_q and lb_q are the upper and lower limits in the q-th dimension, n is the dimension of the variables of the optimization problem, $\beta \in [0,1]$ is an algorithm parameter. Subspace boundaries $\boldsymbol{C}_j$ are modeled as follows:

$$uss_j^q = b_{j,q} + e_d,\ lss_j^q = b_{j,q} - e_d, \tag{5.28}$$

where uss_j^q and lss_j^q are the upper and lower bounds in the q-th dimension of the subspace in accordance. Inside this subspace, h ($h < 4$) new AC are randomly generated, among which the individual with the best value of the fitness function is selected.

Step 7. Learning the knowledge base of the AC.

In this research, the learning method based on evolving artificial neural networks, developed in the research [2], is used to learn the knowledge bases of each AC.

The end of algorithm.

The method of finding solutions using the improved locust swarm algorithm is proposed.

The work of the solution search processing method was simulated according to Steps 1–7. Simulation of the work of the proposed method was carried out in the MathCad 14 software environment (USA). The assessment of elements of the operational situation of the group of troops (forces) was the task to be solved during the simulation.

During the simulation, the following settings of the improved locust algorithm were used: kar=0.75; N=50, gen=1000. The obtained results were averaged over 30 independent runs.

During initialization ($k=0$), the initial population is formed $L^0\left(\{l_1{}^0, l_2{}^0, \ldots, l_N{}^0\}\right)$. The value $\left(\{l_{i1}{}^0, l_{i2}{}^0, \ldots, l_{in}{}^0\}\right)$ of each individual dimension d are distributed randomly and uniformly between the predefined lower initial limit of the parameter lb_d and the upper initial limit of the parameter ub_d:

$$l_{ij}^0 = lb_d + rand \cdot \left(ub_d - lb_d\right),$$

where $i=1\ldots N$, $d=1\ldots n$. In the process of iterative execution of the algorithm, the individual operator $\boldsymbol{A}$ and the swarm operator $\boldsymbol{B}$ are executed until the number of iterations $k = gen$ is reached.

A set of three multivariate test functions were used to test the performance of the improved: Rosenbrock function $f_1(x)$, Sphere function $f_2(x)$ and Ackley function $f_3(x)$. The description of test functions is presented in **Table 5.9**. In the **Table 5.1** n is the dimension of the function; I^n is the interval of variation of variables x_i; x^* is the optimal solution; $f_i(x^*)$ is the minimum value of the function.

Table 5.9 A set of test functions

Test function $f_i(x)$	$x_i \in I^n$	n	**Minimum**
$f_1(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)\times\left(x_i-1\right)^2\right]$	$[-5, 10]^n$	50	$f_1(x^*)=0$, $x^*=(1\ldots1)$
$f_2(x)=\sum_{i=1}^{n}x_i^2$	$[-100, 100]^n$	50	$f_2(x^*)=0$, $x^*=(0\ldots0)$
$f_3(x)=-20\exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)}\right)+20$	$[-32, 32]^n$	50	$F_3(x^*)=0$, $x^*=(0\ldots0)$

The Rosenbrock function $f_1(x)$ has a large, slowly decreasing plateau. The global minimum of the function is inside a parabolic highly elongated surface. The sphere function $f_2(x)$ is unimodal, unlike the Ackley function $f_3(x)$, which is multimodal.

The comparison was based on the following indicators: average best solution, median best solution, and standard deviation from the best solution. The averaged results corresponding to 30 separate launches are given in the **Table 5.10**.

Each cell of the **Table 5.10** shows the average, median solution and standard deviation from the best solution, respectively.

According to **Table 5.2**, the improved locust swarm algorithm provides a gain of 25–28 % compared to the canonical algorithm. The level of significance is 6 % at a value of $T<0.05$ (sums of Wilcoxon ranks for independent samples).

Table 5.10 Comparative evaluation of the proposed algorithm with known ones

Test function $f_1(x)$	Particle swarm algorithm [10]	Differential evolution algorithm [12]	Canonical locust swarm algorithm [23]	Improved locust swarm algorithm
$f_1(x)$	$2.77\cdot10^{-02}$	$2.27\cdot10^{-02}$	$5.47\cdot10^{-04}$	$5.02\cdot10^{-04}$
	$2.66\cdot10^{-02}$	$2.23\cdot10^{-02}$	$4.87\cdot10^{-04}$	$4.1\cdot10^{-04}$
	$5.86\cdot10^{-03}$	$5.03\cdot10^{-03}$	$1.33\cdot10^{-05}$	$1.11\cdot10^{-05}$
$f_2(x)$	$8.55\cdot10^{-03}$	$6.93\cdot10^{-03}$	$9.23\cdot10^{-06}$	$8.82\cdot10^{-06}$
	$3.15\cdot10^{-02}$	$5.53\cdot10^{-05}$	$7.95\cdot10^{-06}$	$7.02\cdot10^{-06}$
	$1.65\cdot10^{-03}$	$1.02\cdot10^{-05}$	$1.17\cdot10^{-06}$	$0.95\cdot10^{-06}$
$f_3(x)$	$3.55\cdot10^{-02}$	$7.02\cdot10^{-04}$	$4.22\cdot10^{-05}$	$3.9\cdot10^{-05}$
	$4.83\cdot10^{-02}$	$7.20\cdot10^{-04}$	$2.65\cdot10^{-05}$	$2.2\cdot10^{-05}$
	$1.2\cdot10^{-03}$	$2.22\cdot10^{-04}$	$3.71\cdot10^{-06}$	$3.1\cdot10^{-06}$

This method will allow:

– to assess the state of the heterogeneous analysis object;

– to determine effective measures to improve management efficiency;

– to increase the speed of assessment of the state of a heterogeneous object of analysis;

– to reduce the use of computing resources of decision support systems.

The limitations of the research are the need to have an initial database on the state of the analysis object, the need to take into account the time delay for collection and proof of information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs when processing various types of data in special purpose systems.

5.4 METHOD FOR FINDING SOLUTIONS USING AN IMPROVED EMPEROR PENGUIN ALGORITHM

The proposed algorithm is an improved emperor penguin algorithm and consists of the following sequence of actions.

Step 1. Input of initial data. At this stage, the initial data available about the object to be analyzed are entered. The existing model of the analysis object is also initialized, the decision matrix D is filled: each column is filled with a subset of F_k.

Step 2. Setting up agents on the search plane.

At this stage, the AIP is issued taking into account the type of uncertainty about the object to be analyzed and the basic model of the object state is initialized [2, 19, 21]. At the same time, the degree of uncertainty can be: full awareness; partial uncertainty and total uncertainty. The above is carried out with the help of appropriate correction coefficients, which are set at the analysis stage.

Step 3. Numbering of AIP in the flock, i, $i \in [0,S]$.

Since penguins are social animals, each member of the flock participates in the general heat exchange: all AIP emit heat, which attracts other members of the AIP colony.

Step 4. Setting the initial speed of AIP and thermal radiation of each AIP.

Initial speed v_0 of each AIP is determined by the following expression:

$$v_i = \left(v_1, v_2 \ldots v_S\right),\ v_i = v_0. \tag{5.29}$$

The initial thermal radiation of each AIP is determined by the expression:

$$T_i = \left(T_1, T_2 \ldots T_S\right), T_i = T_{\max}, \tag{5.30}$$

where T_i is the value of thermal radiation of AIP with number i; $T_{\max}$ is the maximum thermal radiation of AIP.

Step 5. Calculation of the position of each AIP on the total search area and its cost.

The cost of each AIP is calculated according to the formula:

$$Q = A\varepsilon\sigma T_s^4 e^{\mu x}, \tag{5.31}$$

where A is the total surface area of the AIP; ε is the radiative capacity of bird plumage AIP; σ is the Boltzmann's constant; T_i is the AIP body temperature; μ is the coefficient of attenuation of thermal radiation of each AIP, which is calculated taking into account the degree of noise of the initial data about the analysis object, based on the method developed in works [22, 24]; x is the distance to the nearest penguin in meters.

Also, at this stage, the cost of each AIP is compared among themselves. AIPs always move towards an agent that has a low cost (high heat intensity) absorption. The cost of the agent is determined according to the desired function, which accepts the coordinates of the given agent as arguments. In other words, the penguin with the closest position to the global optimum of the function will have the lowest cost, and therefore others will move towards it.

Step 6. An approach (attraction) of AIP to another AIP.

Thus, penguins gather in clusters. However, agents do not move linearly, but spirally. The spiral movement of AIP is described by the formula:

$$\begin{cases} x_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \cdot \cos\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}; \\ y_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \cdot \sin\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}, \end{cases} \tag{5.32}$$

where a is the height of the penguin; b is the thickness of the penguin; Q is the attractiveness of the penguin (calculated from expression (5.31)).

Step 7. Changing the trajectory of AIP movement.

In order not to be limited to a uniform spiral path, a random component is required, which it is possible to introduce using formula (5.32), as follows:

$$\begin{cases} x_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \cdot \cos\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\} + \varphi\varepsilon_i; \\ y_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \cdot \sin\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\} + \varphi\varepsilon_i, \end{cases} \tag{5.33}$$

where φ is the mutation coefficient; ε is a vector of uncertainty information about the state of the analysis object of.

Step 8. Selection of the best individuals from the AIP flock.

At this stage, with the help of the improved genetic algorithm proposed by the authors in the research [25], the best AIPs in the flock are selected based on the indicators of the reduction in heat radiation power of each AIP, the mutation coefficient and the attenuation coefficient.

Step 9. Ranking of the received solutions and their sorting.

After recalculating the AIP position, according to the work (5.32), the mutation coefficient is added, as was shown in the work (5.33). The cost of AIP is again compared, the best result is selected. After each iteration, it is necessary to sort the best solutions and reduce the amount of thermal radiation, mutation rates and attenuation.

Step 10. Learning AR knowledge bases.

In the research, the learning method developed in the work [2] is used to learn the knowledge bases of each AIP, based on artificial neural networks that evolve to change the nature of movement of each AIP, for more accurate analysis results in the future.

Step 11. Determination of the amount of necessary computing resources of intelligent decision-making support system.

In order to prevent looping of calculations on Steps 1–10 of this technique and to increase the efficiency of calculations, the system load is additionally method. When the defined threshold of computational complexity is exceeded, the amount of software and hardware resources that must be additionally involved is determined using the method proposed in the work [37].

The end of algorithm.

To evaluate the effectiveness of the researched methodology, simulations were carried out using some functions of global optimization on the example of a number of test functions, in particular, multi-extreme functions with a complex landscape. As a result of the simulation, sets of input parameters were obtained that ensure the optimal operation of the algorithm under the given conditions (**Table 5.11**).

Table 5.11 The results of the AIP algorithm for test functions

Function	Function dimensionality	Number of AIP	Number of iterations	Result	Average time, sec.
De Jong (global optimum: 0)	2	4	4	0	0
	5	14	18	0	0.1
	10	28	35	0.001	3.17
	30	30	80	0.007	30.42
Rastrigin (global optimum: 0)	2	6	8	0	0
	5	64	40	0	18.23
	10	50	100	0.03	62.5
	30	50	300	0.97	528.4
Hryvnoka (global optimum: 0)	2	6	4	0	0
	5	16	15	0.002	0.17
	10	30	40	0.004	4.77
	30	43	46	0.028	89.38
Ackley (global optimum: 0)	2	6	4	0	0
	5	24	16	0.001	0.15
	10	42	32	0.013	3.24
	30	50	50	0.021	66.73
Bukin (global optimum: 0)	2	8	4	0	0
	5	20	15	0.002	2.08
	10	40	35	0.03	4.16
	30	50	42	0.85	70.4

Analyzing the performance results of the improved algorithm, shown in **Table 5.1**, it can be seen that for functions with a small number of parameters, the algorithm demonstrates its greatest efficiency, however, when the dimension of multi-extremal functions with a complex landscape (such as Rastrigin, Hryvnoka, Bukin functions) increases, there is a slight deviation from global optimum, this deviation can be smoothed out by increasing the number of iterations and agents that affect the duration of the method.

The Rosenbrock function should be noted separately: when the number of parameters increases to more than 10, the AIP shows a rather noticeable deviation from the optimal solution, so to achieve the required accuracy, a serious increase in time costs is required, which makes the method ineffective in this particular case.

Based on the research conducted, it can be said that AIP is more effective for working with functions with a small number of parameters, however, one of the ways to improve the accuracy of the solutions found for multiparameter multimodal functions is to modify or hybridize the method with other algorithms.

The results of the comparative evaluation according to the criterion of efficiency of evaluation with known scientific studies are shown in **Table 5.12**.

Table 5.12 The results of solving the task

Iteration number	Method of branches and boundaries [17]	Genetic algorithm [12]	AIP canonical algorithm [23]	Improved AIP algorithm
N	***T*, s**	***T*, s**	***T*, s**	***T*, s**
5	1.125	1.125	1.1	1.05
10	0.625	0.625	0.611	0.450
15	48.97	58.20	56.2	55.41
20	106.72	44.29	42.75	40.21
30	–0.1790	–0.0018	–0.0003	–0.00007
40	–0.158	–0.070	–0.041	–0.06
50	97.76	–974.30	–3.83	–331.19
100	–133.28	–195.71	–195.15	–198.12
200	7980.89	7207.49	7222.16	7022.85

As can be seen from the **Table 5.11**, the gain of the specified method of finding solutions is from 13 to 17 % according to the criterion of efficiency of data processing.

The limitations of the research are the need to have an initial database on the state of the object of analysis, the need to take into account the time delay for collection and proof of information from intelligence sources.

The proposed approach should be used to solve problems of evaluating complex and dynamic processes characterized by a high degree of complexity.

This research is a further development of researches aimed at developing method principles for increasing the efficiency of processing various types of data, which were published earlier [2, 4–6, 23].

The directions of further research should be aimed at reducing computing costs while processing various types of data in special purpose systems.

CONCLUSIONS

1. A mathematical formulation of the research task was carried out with the help of a pack of wolves. The proposed mathematical formulation of the research task allows to formulate a mechanism for solving the optimization problem using the wolf flock algorithm during the management of hierarchical objects.

2. The method implementation algorithm was defined, which allows:

– to take into account the type of data uncertainty;

– to take into account the available computing resources of the management object state analysis system;

– to increase the efficiency of adaptation of a wolf flock with the help of an improved genetic algorithm, developed in the work [24];

– to carry out an initial display of individuals of a pack of wolves, taking into account the type of uncertainty.

3. Example of using the proposed method was conducted on the example of assessing the state of the operational situation of a group of armies (forces). The conducted simulation showed that the obtained modification of the method of searching by a pack of wolves better solves the task of state analysis and parametric control with different input data to the traveling salesman problem than the classical algorithm of searching by a wolf flock. It also showed better results compared to well-known algorithms for solving this problem, such as the genetic algorithm and the particle swarm algorithm.

The specified example showed an increase in the efficiency of data processing at the level of 23–30 % due to the use of additional improved procedures. The obtained data made it possible to conclude that the time complexity of the algorithm does not exceed the polynomial complexity.

4. The development of a mathematical model of parametric optimization based on the improved firefly algorithm in special purpose information systems was carried out. The specified formalization allows to describe the processes taking place in special purpose information systems while solving the tasks of parametric control of the object state. This approach makes it possible to effectively parallelize the process of searching for an optimal solution, which partially eliminates the problem of prior convergence of the algorithm and to control the search process for finding optimal and quasi-optimal solutions. As a criterion for the effectiveness of the specified method, the efficiency

of decision making regarding the parametric control of the object state of the with the given reliability was chosen. This makes it possible to create a hierarchical description of a complex process by levels of generalization and conduct an appropriate analysis of its state.

5. An algorithm for the implementation of the method is defined, which allows:

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the management object state analysis system;

– to determine the necessary computing resources of the system for operational assessment of the object's condition;

– to carry out accurate training of individuals of a swarm of firefly using the expressions developed in the work [2].

6. An example of the use of the proposed method was conducted on the example of assessing the state of the operational situation of a group of forces (forces). The specified example showed a 17–20 % increase in the efficiency of data processing due to the use of additional improved procedures. The obtained data made it possible to conclude that the time complexity of the algorithm does not exceed the polynomial complexity.

7. An algorithm for the implementation of the method was determined, thanks to additional and improved procedures, which allows:

– to avoid the concentration of AC in the current best positions;

– to reduce the probability of premature convergence of the algorithm;

– to maintain a balance between the speed of convergence of the algorithm and diversification;

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of search for AC;

– to carry out the initial display of AC individuals, taking into account the type of uncertainty;

– to carry out accurate training of AC individuals;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

8. An example of the use of the proposed method was carried out on the example of assessing and forecasting the state of the operational situation of a group of troops (forces). The specified example showed an increase in the efficiency of data processing at the level of 25–28 % due to the

use of additional improved procedures for adding correction coefficients for uncertainty and noisy data, as well as training of the AC.

9. An algorithm for the implementation of the method was determined, thanks to additional and improved procedures, which allows:

– to take into account the type of uncertainty and noisy data;

– to take into account the available computing resources of the state analysis system of the analysis object;

– to take into account the priority of the AIP search;

– to carry out the initial exhibition of AIP individuals taking into account the type of uncertainty;

– to carry out accurate training of AIP individuals;

– to determine the best AR individuals using a genetic algorithm;

– to conduct a local and global search taking into account the degree of noise of the data on the state of the analysis object;

– to conduct training of knowledge bases, which is carried out by training the synaptic weights of the artificial neural network, the type and parameters of the membership function, as well as the architecture of individual elements and the architecture of the artificial neural network as a whole;

– to be used as a universal tool for solving the task of analyzing the state of analysis objects due to the hierarchical description of analysis objects;

– to check the adequacy of the obtained results;

– to avoid the problem of local extremum.

10. An example of the use of the proposed method was carried out on the example of assessing and forecasting the state of the operational situation of a group of armies (forces). The specified example showed an increase in the efficiency of data processing at the level of 13–17 % due to the use of additional improved procedures of adding correction coefficients for uncertainty and noisy data, AIP selection, as well as AIP training.

REFERENCES

1. Shyshatskyi, A. V., Bashkyrov, O. M., Kostyna, O. M. (2015). Rozvytok intehrovanykh system zviazku ta peredachi danykh dlia potreb Zbroinykh Syl. Ozbroiennia ta viiskova tekhnika, 1 (5), 35–40.
2. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskyi, R., Repilo, I. et al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 37–47. https://doi.org/10.15587/1729-4061.2020.203301
3. Sova, O., Shyshatskyi, A., Salnikova, O., Zhuk, O., Trotsko, O., Hrokholskyi, Y. (2021). Development of a method for assessment and forecasting of the radio electronic environment. EUREKA: Physics and Engineering, 4, 30–40. https://doi.org/10.21303/2461-4262.2021.001940

4. Pievtsov, H., Turinskyi, O., Zhyvotovskyi, R., Sova, O., Zvieriev, O., Lanetskii, B., Shyshatskyi, A. (2020). Development of an advanced method of finding solutions for neuro-fuzzy expert systems of analysis of the radioelectronic situation. EUREKA: Physics and Engineering, 4, 78–89. https://doi.org/10.21303/2461-4262.2020.001353
5. Zuiev, P., Zhyvotovskyi, R., Zvieriev, O., Hatsenko, S., Kuprii, V., Nakonechnyi, O. et al. (2020). Development of complex methodology of processing heterogeneous data in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 14–23. https://doi.org/10.15587/1729-4061.2020.208554
6. Shyshatskyi, A., Zvieriev, O., Salnikova, O., Demchenko, Ye., Trotsko, O., Neroznak, Ye. (2020). Complex Methods of Processing Different Data in Intellectual Systems for Decision Support System. International Journal of Advanced Trends in Computer Science and Engineering, 9 (4), 5583–5590. https://doi.org/10.30534/ijatcse/2020/206942020
7. Yeromina, N., Kurban, V., Mykus, S., Peredrii, O., Voloshchenko, O., Kosenko, V. et al. (2021). The Creation of the Database for Mobile Robots Navigation under the Conditions of Flexible Change of Flight Assignment. International Journal of Emerging Technology and Advanced Engineering, 11 (5), 37–44. https://doi.org/10.46338/ijetae0521_05
8. Rotshtein, A. P. (1999). Intellektualnye tekhnologii identifikatcii: nechetkie mnozhestva, geneticheskie algoritmy, neironnye seti. Vinnitca: UNIVERSUM, 320.
9. Alpeeva, E. A., Volkova, I. I. (2019). The use of fuzzy cognitive maps in the development of an experimental model of automation of production accounting of material flows. Russian Journal of Industrial Economics, 12 (1), 97–106. https://doi.org/10.17073/2072-1633-2019-1-97-106
10. Zagranovskaia, A. V., Eissner, Iu. N. (2017). Simulation scenarios of the economic situation based on fuzzy cognitive maps. Modern economics: problems and solutions, 10 (94), 33–47. https://doi.org/10.17308/meps.2017.10/1754
11. Simankov, V. S., Putiato, M. M. (2013). Issledovanie metodov kognitivnogo analiza. Sistemnyi analiz, upravlenie i obrabotka informatcii, 13, 31–35.
12. Ko, Y.-C., Fujita, H. (2019). An evidential analytics for buried information in big data samples: Case study of semiconductor manufacturing. Information Sciences, 486, 190–203. https://doi.org/10.1016/j.ins.2019.01.079
13. Ramaji, I. J., Memari, A. M. (2018). Interpretation of structural analytical models from the coordination view in building information models. Automation in Construction, 90, 117–133. https://doi.org/10.1016/j.autcon.2018.02.025
14. Pérez-González, C. J., Colebrook, M., Roda-García, J. L., Rosa-Remedios, C. B. (2019). Developing a data analytics platform to support decision making in emergency and security management. Expert Systems with Applications, 120, 167–184. https://doi.org/10.1016/j.eswa.2018.11.023
15. Chen, H. (2018). Evaluation of Personalized Service Level for Library Information Management Based on Fuzzy Analytic Hierarchy Process. Procedia Computer Science, 131, 952–958. https://doi.org/10.1016/j.procs.2018.04.233

16. Chan, H. K., Sun, X., Chung, S.-H. (2019). When should fuzzy analytic hierarchy process be used instead of analytic hierarchy process? Decision Support Systems, 125, 113114. https://doi.org/10.1016/j.dss.2019.113114
17. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. Future Generation Computer Systems, 91, 620–633. https://doi.org/10.1016/j.future.2018.06.046
18. Gödri, I., Kardos, C., Pfeiffer, A., Váncza, J. (2019). Data analytics-based decision support workflow for high-mix low-volume production systems. CIRP Annals, 68 (1), 471–474. https://doi.org/10.1016/j.cirp.2019.04.001
19. Harding, J. L. (2013). Data quality in the integration and analysis of data from multiple sources: some research challenges. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-2/W1, 59–63. https://doi.org/10.5194/isprsarchives-xl-2-w1-59-2013
20. Kosko, B. (1986). Fuzzy cognitive maps. International Journal of Man-Machine Studies, 24 (1), 65–75. https://doi.org/10.1016/s0020-7373(86)80040-2
21. Gorelova, G. V. (2013). Cognitive approach to simulation of large systems. Izvestiia IuFU. Tekhnicheskie nauki, 3, 239–250.
22. Orouskhani, M., Orouskhani, Y., Mansouri, M., Teshnehlab, M. (2013). A Novel Cat Swarm Optimization Algorithm for Unconstrained Optimization Problems. International Journal of Information Technology and Computer Science, 5 (11), 32–41. https://doi.org/10.5815/ijitcs.2013.11.04
23. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. https://doi.org/10.15587/1729-4061.2019.180197
24. Mahdi, Q. A., Shyshatskyi, A., Prokopenko, Y., Ivakhnenko, T., Kupriyenko, D., Golian, V. et al. (2021). Development of estimation and forecasting method in intelligent decision support systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 51–62. https://doi.org/10.15587/1729-4061.2021.232718
25. Emelianov, V. V., Kureichik, V. V., Kureichik, V. M., Emelianov, V. V. (2003). Teoriia i praktika evoliutcionnogo modelirovaniia. Moscow: Fizmatlit, 432.
26. Gorokhovatsky, V., Stiahlyk, N., Tsarevska, V. (2021). Combination method of accelerated metric data search in image classification problems. Advanced Information Systems, 5 (3), 5–12. https://doi.org/10.20998/2522-9052.2021.3.01
27. Levashenko, V., Liashenko, O., Kuchuk, H. (2020). Building Decision Support Systems based on Fuzzy Data. Advanced Information Systems, 4 (4), 48–56. https://doi.org/10.20998/2522-9052.2020.4.07
28. Meleshko, Y., Drieiev, O., Drieieva, H. (2020). Method of identification bot profiles based on neural networks in recommendation systems. Advanced Information Systems, 4 (2), 24–28. https://doi.org/10.20998/2522-9052.2020.2.05

29. Kuchuk, N., Merlak, V., Skorodelov, V. (2020). A method of reducing access time to poorly structured data. Advanced Information Systems, 4 (1), 97–102. https://doi.org/10.20998/2522-9052.2020.1.14
30. Shyshatskyi, A., Tiurnikov, M., Suhak, S., Bondar, O., Melnyk, A., Bokhno, T., Lyashenko, A. (2020). Method of assessment of the efficiency of the communication of operational troop grouping system. Advanced Information Systems, 4 (1), 107–112. https://doi.org/10.20998/2522-9052.2020.1.16
31. Raskin, L., Sira, O. (2016). Method of solving fuzzy problems of mathematical programming. Eastern-European Journal of Enterprise Technologies, 5 (4 (83)), 23–28. https://doi.org/10.15587/1729-4061.2016.81292
32. Lytvyn, V., Vysotska, V., Pukach, P., Brodyak, O., Ugryn, D. (2017). Development of a method for determining the keywords in the slavic language texts based on the technology of web mining. Eastern-European Journal of Enterprise Technologies, 2 (2 (86)), 14–23. https://doi.org/10.15587/1729-4061.2017.98750
33. Stepanenko, A., Oliinyk, A., Deineha, L., Zaiko, T. (2018). Development of the method for decomposition of superpositions of unknown pulsed signals using the secondorder adaptive spectral analysis. Eastern-European Journal of Enterprise Technologies, 2 (9 (92)), 48–54. https://doi.org/10.15587/1729-4061.2018.126578
34. Gorbenko, I., Ponomar, V. (2017). Examining a possibility to use and the benefits of post-quantum algorithms dependent on the conditions of their application. Eastern-European Journal of Enterprise Technologies, 2 (9 (86)), 21–32. https://doi.org/10.15587/1729-4061.2017.96321
35. Lovska, A. A. (2015). Peculiarities of computer modeling of strength of body bearing construction of gondola car during transportation by ferry-bridge. Metallurgical and Mining Industry, 1, 49–54.
36. Lovska, A., Fomin, O. (2020). A New fastener to ensure the reliability of a passenger car body on a train ferry. Acta Polytechnica, 60 (6), 478–485. https://doi.org/10.14311/ap.2020.60.0478
37. Koval, M., Sova, O., Orlov, O., Shyshatskyi, A., Artabaiev, Y., Shknai, O., Veretnov, A., Koshlan, O., Zhyvylo, Y., Zhyvylo, I. (2022). Improvement of complex resource management of special-purpose communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 34–44. https://doi.org/10.15587/1729-4061.2022.266009

Edited by
Andrii Shyshatskyi

INFORMATION AND CONTROL SYSTEMS: MODELLING AND OPTIMIZATIONS

Olexander Litvinenko, Svitlana Kashkevich, Andrii Shyshatskyi, Oksana Dmytriieva, Serhii Neronov, Ganna Plekhova, Yevhen Zhyvylo, Nina Kuchuk, Oleksii Kuvshynov, Oleksandr Yefymenko, Andrii Veretnov, Heorhii Kuchuk, Viacheslav Davydov, Yurii Beketov, Illia Dmytriiev, Inna Shevchenko, Oleksandr Lytvynenko, Lyubov Shabanova-Kushnarenko, Nataliia Apenko

Collective monograph

Technical editor I. Prudius
Desktop publishing T. Serhiienko

TECHNOLOGY CENTER PC®
Published in August 2024
Enlisting the subject of publishing No. 4452 – 10.12.2012
Address: Shatylova dacha str., 4, Kharkiv, Ukraine, 61165